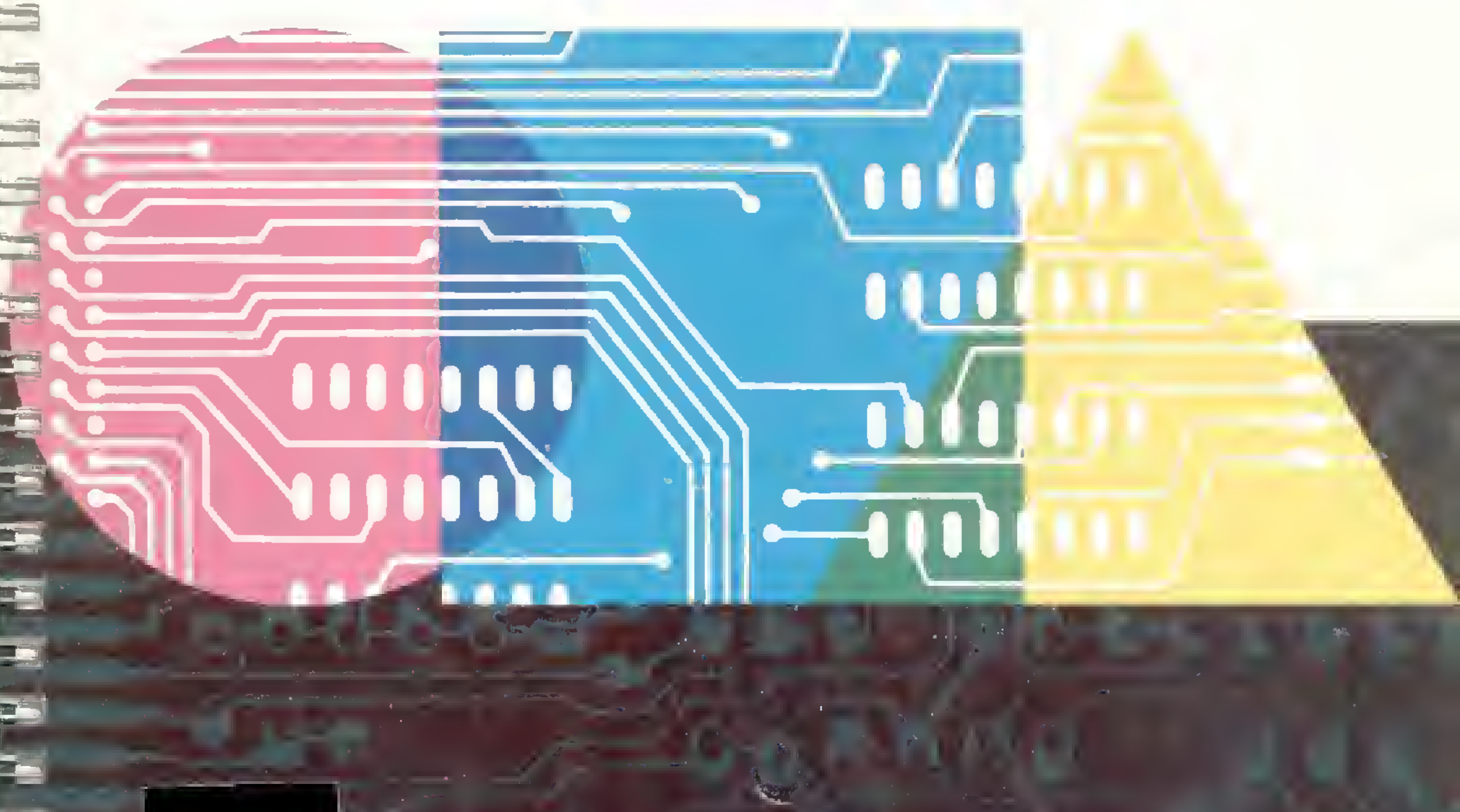
21741

CIRCUIT DESIGN PROGRAMS FOR THE TRS-80®

HOWARD M. BERLIN



BLACKSBURG CONTINUING EDUCATION SERIES™
edited by Titus, Titus & Larsen

Circuit Design Programs for the TRS-80®

by
Howard M. Berlin

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

FIRST EDITION
FIRST PRINTING—1980

International Standard Book Number: 0-672-21741-4
Library of Congress Catalog Card Number: 80-52227

Printed in the United States of America.

Preface

The great mathematician Gottfried Wilhelm Leibniz perhaps stated it best when he said: "It is unworthy of excellent men to lose hours like slaves in the labor of calculation." Before there were computers, virtually every aspect of electronic circuit design and analysis was performed either by hand, or with a slide rule. When done by hand, there was always a good chance that a computational error would exist. With a slide rule, the preciseness of the result left much to be desired.

With the great strides made in the integrated circuit revolution, there now exists a number of relatively inexpensive "personal-type" digital computers, such as Radio Shack's Model TRS-80®, which can efficiently perform virtually any type of computation required for the design and analysis of electronic networks. Consequently, it is the goal of this book to present a variety of useful BASIC language programs which will greatly simplify the design and analysis of commonly encountered circuit problems.

All of the programs listed in this book were primarily written for a Radio Shack TRS-80 system having Level-II BASIC with at least 16K of random access memory. Level-II BASIC offers many advantages over the limited LEVEL-I set of commands and statements, particularly the mathematical functions, many of which are used repeatedly in the programs described in this book. However, many of these routines can be easily adapted to run on other computer systems, such as the PET and APPLE II. For example, the TRS-80 "SET" function is the same as the APPLE II "PLOT" function. The program listings and the actual output results, as they would normally appear on the video display, were printed using an IBM Selectric® printer instead of the dot-matrix font, to make the reading easier.

In eight chapters, a variety of programs are presented that enable the user to solve a myriad of problems relating to plotting and the simple statistical verification of experimental data; rms and average values, as well as the Fourier series of a periodic waveform; inverse Laplace, Fourier, and other network transforms; design of matching pads, attenuators, active filters, heat sinks, integrated circuit timer, zener diode regulator, and bipolar transistor circuits; the solution of simultaneous equations with real or complex coefficients, damped oscillations, and polynomial roots. Some of the programs, such as plot routines, are written as subroutines so that they may be used as part of some of your own programs.

All output is displayed on the video screen, but most programs can be easily modified so that numerical results are handled by a line printer. The least squares regression program has the capability to store data on cassette tape. For those of you who have disk drives, this routine is easily modified to place the created data file on disk. Each program is assigned a name, which may be used as the file specification for those of you storing programs on disk media.

Several of the programs in this book were modified from programs originally written in FORTRAN, which were part of extensive subroutine packages. Others were adapted from BASIC programs that have been published elsewhere; however, the algorithms are standard. The mathematical theory for a particular circuit design or numerical technique used was intentionally kept to a minimum, since this book was not meant to replace traditional texts on circuit theory or numerical analysis and methods. However, there are short descriptions of the programs, along with computed examples and applicable schematic dia-

grams, which are used to aid in both the explanation and verification of results against your own copies of the programs.

Appendix A lists all the programs discussed in this book along with their approximate memory capacities. In addition, Appendix B lists the standard values for resistors and capacitors, which are helpful for a number of the programs.

There are several computers, without whose help this book would still be on the proverbial drawing board. In particular, I am especially indebted to my own TRS-80 computer, aided by the Electric Pencil® word processor and an IBM Selectric® typewriter for its efficient editing and typing of the manuscript and program listings before it went to the publisher. In addition, special thanks goes to a certain Sperry-Univac model V77-600 16-bit computer-based interactive graphics system equipped with a Calcomp model 1055 plotter driven by a Sperry-Univac model 620/f-100 computer, which drew all the drawings in this book, as well as the ones for my earlier *555 Timer Applications Sourcebook, With Experiments.*

Although this is my seventh book in the Blacksburg Continuing Education Series, I feel that I have had the most fun "writing" this one, even though my computers will loudly claim that they did all the work!

HOWARD M. BERLIN

Contents

CHAPTER 1

About the Programs 7

Basic Facts—Using a Line Printer—Merging a Program With a Subroutine

CHAPTER 2

Plotting Graphs. 9

Histograms—Cartesian Plots—Semilogarithmic Plots—Log-Log Plots—Polar Plots—Plotting With a Line Printer

CHAPTER 3

The Analysis of Signals 33

The Average and RMS Values of a Waveform—Fourier Series—Fourier Transform—Analysis of Damped Oscillations

CHAPTER 4

Basic Statistics and the Analysis of Data 49

Least Squares Regression Curves—Sample Statistics

CHAPTER 5

Networks and Transforms 63

Roots of Polynomials—The Inverse Laplace Transform—Impedance Matching Pads—Pi-Tee (Delta-Wye) Transforms—Mesh Current Network Analysis

CHAPTER 6

Active Filter Design 87

Low-Pass and High-Pass Filters—The State-Variable Filter—A Bandpass Filter for Q Less Than 10—A Bandpass Filter for Q Less Than 50—Staggered-Tuned Butterworth Bandpass Filters—The Notch Filter—A Wideband Filter

CHAPTER 7

SOLID-STATE DEVICES 113

Zener Diode Voltage Regulator Design—555 Timer Design—Transistors—Heat Sink Design

CHAPTER 8

ADDITIONAL ROUTINES 125

Four-Quadrant Arctangent Function—Complex Number Math—Minimum and Maximum Values of an Array

APPENDIX A

PROGRAMS AND SUBROUTINES 133

APPENDIX B

STANDARD RESISTOR AND CAPACITOR VALUES 135

INDEX 137

1

About the Programs

BASIC FACTS

The programs described in this book were written for a Level-II system having at least 16K of read/write memory. In addition, it is assumed that the user does not have either line printer or disk peripherals. When in Level-II BASIC, all programs are loaded from cassette tape using the CLOAD command. By the same token, data files created by a program are stored on tape. However, if disk and/or printer peripherals are available, many of the programs can be easily modified to either output results to a line printer or store/retrieve data to/from disk. If the programs and data files are stored on disk, then there should be at least 32K of read/write memory. This is because several programs are longer than 5K, as the disk operating system and disk BASIC occupy approximately 10K of memory when loaded.

All computations are performed using single precision arithmetic, so that most numerical results will be displayed with a maximum of six digits. For the majority of various designs, this has shown to be sufficient. In a number of cases, numerical calculations are provided with enough decimal places to make the results significant. As an example, resistance values calculated in kilohms are usually displayed with two decimal places.

USING A LINE PRINTER

Suppose a line printer is available. If the program is relatively short and does not contain too many PRINT statements, one can easily use the Level-II EDIT commands to change all PRINT statements to LPRINT. On the other hand, suppose the program is quite long, which is usually the case, changing all the necessary statements can be time consuming. In addition, Murphy's law virtually assures us that some of them would probably be missed.

By using the following two POKE commands:

```
POKE 16414,141
POKE 16415,5
```

placed in the program before the first PRINT statement, all items will be directed to the line printer instead of them appearing on the video display.

On the other hand, the following two statements:

```
POKE 16414,88
POKE 16415,4
```

must be inserted before your program ends in order to reverse the process. If this is not done, both the READY message and the > prompt cursor will be printed on the line printer instead of appearing on the video display. In addition, the system will have to be re-booted, since one will now be unable to return to having messages displayed on the screen.

When INPUT statements are located between these two pair of POKE statements, parallel-output line printers, such as Radio Shack's TRS-80 Line Printer I (No. 26-1152), which is equivalent to the Centronics 779, will LPRINT the response only after the ENTER key is pressed. Consequently, one will not be aware of a typing error before pressing the ENTER key. On the other hand, RS-232 serial printers, such as teletypes and modified IBM Selectric typewriters simultaneously print each character as it is entered from the keyboard.

MERGING A PROGRAM WITH A SUBROUTINE

A number of programs in this book were purposely written as subroutines so that they may be made part of your own application programs. These include a variety of plotting routines, and mathematical functions, such as a four-quadrant arctangent function. Rather than retype the subroutine each time it is needed, one can easily merge, or combine, the subroutine which is already stored on tape with the main program which is presently in memory.

To merge a subroutine already stored on tape with a main program in memory, perform the following steps:

1. The line numbers of the program stored on tape must be larger than the highest line number of the main program. The subroutines in this book start with line numbers 5000 and higher, and no two subroutines have the same inclusive line numbers.
2. While in BASIC II's COMMAND mode, determine the values stored in locations 16633 and 16634 by typing:

   ```
   PRINT PEEK(16633),PEEK(18634)
   ```

   and then pressing the ENTER key.
3. If the value stored in location 16633 is equal to or greater than 2, then execute the following two statements in the COMMAND mode:

   ```
   POKE 16548,PEEK(16633)-2
   POKE 16549,PEEK(16634)
   ```

   and then use the CLOAD command to load the program stored on cassette tape from the tape recorder, after which the following two statements are executed:

   ```
   POKE 16548,233
   POKE 16549,66
   ```

   The merged program can then be saved on tape using the CSAVE command.
4. If the value stored in location 16633 is less than 2, then execute the following two statements:

   ```
   POKE 16548,PEEK(16633)+254
   POKE 16549,PEEK(16834)-1
   ```

   and then use the CLOAD command to load the program stored on cassette tape from the tape recorder, after which the following two statements are executed:

   ```
   POKE 16548,233
   POKE 16549,66
   ```

   The merged program can then be saved on tape using the CSAVE command.

For those using one or more disk drives, programs can be merged while in DOS by using the APPEND command (TRSDOS version 2.2 or later), in which case both programs must have been previously saved using the "A" option (ASCII format).

As an example, the ARCTAN subroutine, starting with line number 6000, is located on disk drive 1, and the main program called MAINPRGM with which the ARCTAN program is to be merged is located on drive 0. Remember, both programs must have been saved in ASCII format and that the highest line number of MAINPRGM must be less than 6000. To merge these two programs, execute the DOS command:

```
APPEND ARCTAN:1 TO MAINPRGM:0
```

in which case MAINPRGM will now include the lines of ARCTAN. The ARCTAN file remains unchanged.

2

Plotting Graphs

Perhaps one of the best methods for describing mathematical functions or experimental results is to plot them in the form of a graph. Remember, a good picture is worth ten thousand words. This chapter illustrates how it is possible to graph functions or discrete data points on your video display in the form of histograms, Cartesian, semilogarithmic, log-log, or polar plots. Most of these programs are primarily written as subroutines so that they may be easily used in conjunction with your own special programs. In addition, programs are included which allow plotting of various types of graphs on a line printer. Once stored on tape or disk, the subroutines can then be easily merged with other main programs as needed.

HISTOGRAMS

One of the simplest types of graphs is the histogram, or bar graph, which is usually used to display frequency distributions. Fig. 2-1 shows the listing of the PHISTGM subroutine used to generate a histogram on the video display. Required input variables are: the number of data points N, the horizontal and vertical values X(I) and Y(I), respectively, for each point, and the graph title T$. Fig. 2-2 illustrates a simple driver program that can be used with the PHISTGM subroutine. The program assumes that each X-axis value has only a single corresponding Y-axis value. Consequently, only a maximum of 101 individual bars may be plotted. The DIM statement in line 101 must be included, as the coordinate values are stored before they are plotted on the display.

Example 2-1

Plot a histogram of the result of measuring the values of a sample of 55, 10% resistors, all of which are supposed to have the same nominal value, 100 ohms. As a result of our sampling, we find that the values vary, with several resistors having like values, as listed in Table 2-1.

Table 2-1. Data for Example 2-1

Point No.	Value	Frequency
1	90	1
2	92	1
3	93	2
4	94	4
5	95	3
6	96	4
7	98	7
8	99	8
9	100	11
10	101	4
11	103	2
12	104	3
13	106	3
14	110	2
		Total = 55

Using the above data, Fig. 2-3A shows the results of executing the main program of Fig. 2-2 and the PHISTGM subroutine (Fig. 2-1) as one program (refer to Chapter 1 for how a program stored on tape can be merged with another program already in memory). After all the 14 individual points and the graph title are entered, the display informs us as to the current ranges of the two axes. The program then asks us how we want the axes scaled, such as having the horizontal axis range from 80 to 120 while the vertical axis varies from 0 to 15. Because of the method in which the axes are scaled, it is best to have the vertical axis (i.e., the "frequency" axis) incremented by a number that is easily divisible by five, since there are five major divisions, or tic

```
5000 ' HISTOGRAM SUBROUTINE - PHISTGM
5001 ' X(I)=HORIZONTAL AXIS    Y(I)=VERTICAL AXIS
5002 ' N=NUMBER OF DATA POINTS    T$=GRAPH TITLE
5003 Y1=Y(1):Y2=Y(1):X1=X(1):X2=X(1)
5004 FOR I=2TON
5005 IF(Y1-Y(I))<=0 THEN 5008  ELSE 5006
5006 Y1=Y(I)
5007 GOTO 5010
5008 IF(Y2-Y(I))<0 THEN 5009  ELSE 5010
5009 Y2=Y(I)
5010 IF(X1-X(I))<=0 THEN 5013  ELSE 5011
5011 X1=X(I)
5012 GOTO 5015
5013 IF(X2-X(I))<0 THEN 5014  ELSE 5015
5014 X2=X(I)
5015 NEXTI
5016 PRINT"MIN - MAX X VALUES ARE: ";X1,X2
5017 PRINT"MIN - MAX Y VALUES ARE: ";Y1,Y2
5018 INPUT"MIN,MAX X-AXIS SCALE: ";X1,X2
5019 INPUT"MIN-MAX Y-AXIS SCALE: ";Y1,Y2
5020 XD=X2-X1:YD=Y2-Y1
5021 CLS
5022 PRINT@0,Y2:PRINT@128,Y1+(YD*4/5):PRINT@256,Y1+(YD*3/5):PRINT@448,Y1+(YD*2/5)
5023 PRINT@576,Y1+(YD/5):PRINT@704,Y1
5024 FOR Y=0TO35
5025 SET(13,Y):NEXTY
5026 FOR X=15TO115:SET(X,35):NEXTX
5027 FOR X=15TO115 STEP10:SET(X,36):NEXTX
5028 FOR Y=0TO35 STEP 7
5029 SET(12,Y)
5030 NEXTY
5031 PRINT@838,X1:PRINT@848,X1+(XD/5):PRINT@858,X1+(XD*2/5)
5032 PRINT@868,X1+(XD*3/5):PRINT@878,X1+(XD*4/5):PRINT@888,X2
5033 PRINT@916,T$
5034 FOR I=1TON
5035 X=115-((X2-X(I))*100/XD):Y=35-((Y(I)-Y1)*35/YD)
5036 IF X>116 THEN 5040  ELSE 5037
5037 YT=Y
5038 FOR Y=YT TO 35:SET(X,Y):NEXTY
5039 NEXTI
5040 INPUT"HIT <ENTER> TO CONTINUE";FF
5041 CLS:INPUT"ANY CHANGES IN SCALE FACTORS (YES/NO) ";A$
5042 IF A$="YES" THEN 5016  ELSE 5043
5043 RETURN
```

Fig. 2-1. Listing for PHISTGM subroutine.

```
100 'MAIN PROGRAM FOR PHISTGM SUBROUTINE
101 CLS:DIM X(101),Y(101)
102 INPUT"NUMBER OF DATA POINTS TO BE PLOTTED ";N
103 FOR I=1 TO N
104 PRINT"POINT";I;
105 INPUT"X, Y ";X(I),Y(I)
106 NEXT I
107 INPUT"TITLE FOR GRAPH ";T$
108 GOSUB 5000                    'PHISTGM SUBROUTINE
109 END
```

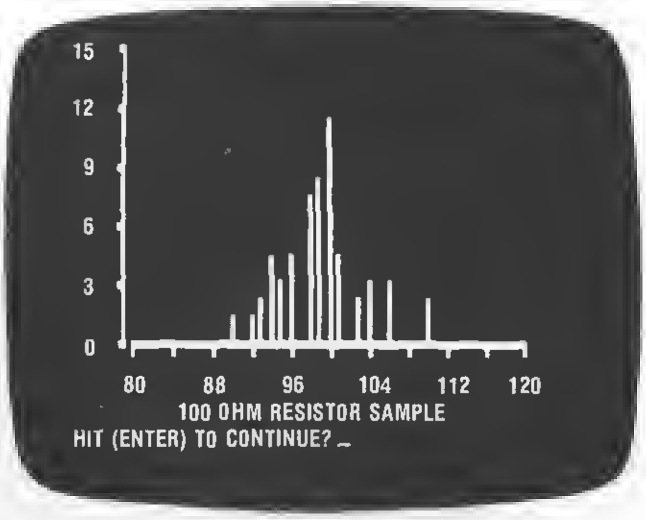
Fig. 2-2. Simple main program for PHISTGM.

marks, on this axis. When the ENTER key is pressed, the resultant histogram is drawn on the video display, as shown in Fig. 2-3B.

The histogram then remains on the screen until the ENTER key is pressed, after which we are asked whether or not we are satisfied with how the axes were scaled, as shown by the new entries in Fig. 2-4A and the resultant graph in Fig.

```
NUMBER OF DATA POINTS TO BE PLOTTED ? 14
POINT 1 X, Y ? 90,1
POINT 2 X, Y ? 92,1
POINT 3 X, Y ? 93,2
POINT 4 X, Y ? 94,4
POINT 5 X, Y ? 95,3
POINT 6 X, Y ? 96,4
POINT 7 X, Y ? 98,7
POINT 8 X, Y ? 99,8
POINT 9 X, Y ? 100,11
POINT 10 X, Y ? 101,4
POINT 11 X, Y ? 103,2
POINT 12 X, Y ? 104,3
POINT 13 X, Y ? 106,3
POINT 14 X, Y ? 110,2

TITLE FOR GRAPH ? 100 OHM RESISTOR SAMPLE

MIN - MAX X VALUES ARE:  90
MIN - MAX Y VALUES ARE:  1
MIN,MAX X-AXIS SCALE: ? 80,120
MIN,MAX Y-AXIS SCALE: ? 0,15
```

(A) Output results.



(B) Video plot.

Fig. 2-3. Plot of histogram of resistors.

2-4B. The process of changing the axis scales can be repeated as many times as desired.

For those using TRSDOS, line 5040 can be changed to:

```
LINEINPUT" ";F$
```

so that a "-" prompt appears in the lower left-hand corner instead of the "HIT <ENTER> TO CONTINUE?" prompt.

CARTESIAN PLOTS

The program for generating a Cartesian, or rectangular X-Y coordinate plot is similar to the histogram. The PCARTXY subroutine is shown in Fig. 2-5, and can be used with the plot driver program of Fig. 2-2, except that the GOSUB 5000 statement in line 108 is now replaced with GO-SUB 5050.

(A) Entries for new scales.

```
ANY CHANGES IN SCALE FACTORS (YES/NO) ? YES
MIN - MAX X VALUES ARE:  80             120
MIN - MAX Y VALUES ARE:  0              15
MIN,MAX X-AXIS SCALE: ? 50,150
MIN,MAX Y-AXIS SCALE: ? 0,15
```

(B) Video plot.

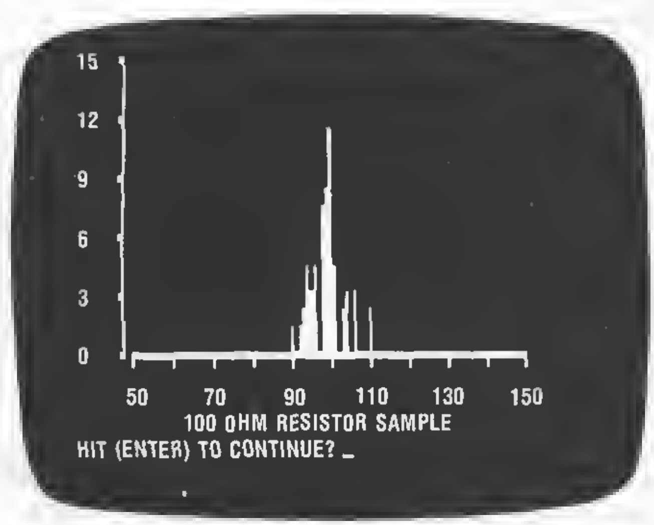


Fig. 2-4. Histogram of resistors with new scales.

```
5050 ' X-Y PLOT SUBROUTINE (PCARTXY)
5051 ' X(I)=HORIZONTAL AXIS
5052 ' Y(I)=VERTICAL AXIS
5053 ' N=NUMBER OF DATA POINTS
5054 ' T$=GRAPH TITLE
5055 ' DETERMINE MIN & MAX VALUES OF X & Y
5056 Y1=Y(1):Y2=Y(1):X1=X(1):X2=X(1)
5057 FOR I=2TON
5058 IF(Y1-Y(I))<=0 THEN 5061 ELSE 5059
5059 Y1=Y(I)
5060 GOTO 5063
5061 IF(Y2-Y(I))<0 THEN 5062 ELSE 5063
5062 Y2=Y(I)
5063 IF(X1-X(I))<=0 THEN 5066 ELSE 5064
5064 X1=X(I)
5065 GOTO 5068
5066 IF(X2-X(I))<0 THEN 5067 ELSE 5068
5067 X2=X(I)
5068 NEXTI
5069 PRINT"CURRENT MIN - MAX X VALUES ARE: ";X1,X2
5070 PRINT"CURRENT MIN - MAX Y VALUES ARE: ";Y1,Y2
5071 INPUT"DESIRED MIN,MAX X-AXIS SCALE: ";X1,X2
5072 INPUT"DESIRED MIN,MAX Y-AXIS SCALE: ";Y1,Y2
5073 XD=X2-X1:YD=Y2-Y1
5074 ' PLOT X-Y AXES AND SCALES
5075 CLS
5076 PRINT@0,Y2:PRINT@128,Y1+(YD*4/5):PRINT@256,Y1+(YD*3/5):PRINT@448,Y1+(YD*2/5)
5077 PRINT@576,Y1+(YD/5):PRINT@704,Y1
5078 FOR Y=0TO35
5079 SET(13,Y):NEXTY
5080 FOR X=15TO115:SET(X,37):NEXTX
5081 FOR X=15TO115 STEP10:SET(X,38):NEXTX
5082 FOR Y=0TO35 STEP 7
5083 SET(12,Y)
5084 NEXTY
5085 PRINT@838,X1:PRINT@848,X1+(XD/5):PRINT@858,X1+(XD*2/5)
5086 PRINT@868,X1+(XD*3/5):PRINT@878,X1+(XD*4/5):PRINT@888,X2
5087 PRINT@916,T$
5088 FOR I=1TON
5089 ' PLOT POINTS
5090 X=115-((X2-X(I))*100/XD):Y=35-((Y(I)-Y1)*35/YD)
5091 IFX>115 THEN 5093 ELSE 5092
5092 SET(X,Y):NEXTI
5093 INPUT"HIT <ENTER> TO CONTINUE ";FF:CLS
5094 INPUT"ANY CHANGES IN SCALE FACTORS (YES/NO) ";A$
5095 IF A$="YES" THEN 5069 ELSE 5096
5096 RETURN
```

Fig. 2-5. Listing for PCARTXY subroutine.

Example 2-2

Graph the following 6 discrete (X,Y) points: (0,0.032), (0.02,0.135), (0.04,0.187), (0.06,0.268), (0.08,0.359), and (0.1,0.435). Fig. 2-6 shows the results after this data is entered. The X-axis varies from 0 to 0.1 as Y ranges from 0.032 to 0.435, based on our data points. The X-axis was then scaled from 0 to 0.1 while the Y-axis was conveniently set to range from 0 to 0.45.

On the other hand, suppose we are interested in graphing continuous mathematical functions, instead of discrete data points. The following example illustrates how this is accomplished.

Example 2-3

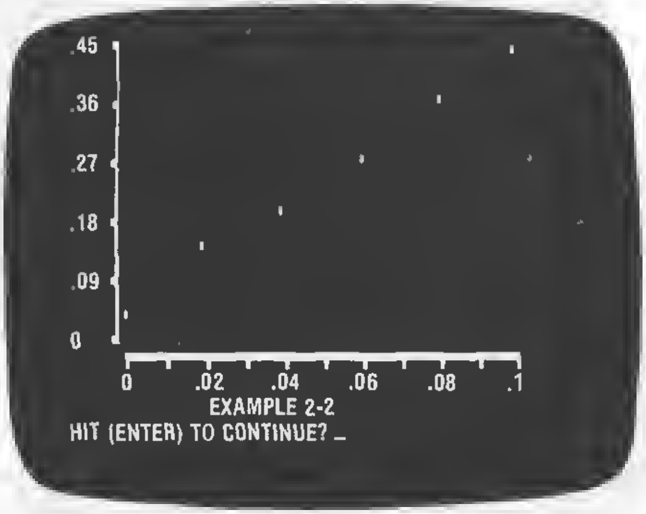
Suppose the transient response of a given network has the known time varying solution:

$$y(t) = 2.3te^{-5t}$$

Fig. 2-7 lists the main program used to plot this function, which is defined in line 106 in terms of the variables y and time, t (the independent variable). Line 107 transforms these two variables to

```
NUMBER OF DATA POINTS TO BE PLOTTED ? 6
POINT 1 X, Y ? 0,0.032
POINT 2 X, Y ? 0.02,0.135
POINT 3 X, Y ? 0.04,0.187
POINT 4 X, Y ? 0.06,0.268
POINT 5 X, Y ? 0.08,0.359
POINT 6 X, Y ? 0.1,0.435

TITLE FOR GRAPH ? EXAMPLE 2-2 DATA
CURRENT MIN - MAX X VALUES ARE:  0                    .1
CURRENT MIN - MAX Y VALUES ARE:  .032                      .435
DESIRED MIN,MAX X-AXIS SCALE: ? 0,.1
DESIRED MIN,MAX Y-AXIS SCALE: ? 0,.45
```

(A) Output results.



(B) Video plot.

Fig. 2-6. Number of data points plotted.

the variable arrays Y(I) and X(I), respectively, which are required by the PCARTXY subroutine. Since most functions are single-valued, there will be 101 continuous points, which are recognized by the DIM statement in line 101. When this main program is run, we are first asked to select the minimum and maximum values of time over which this function is to be plotted, after which we are asked for a suitable title for our graph.

For this example, we are interested in how this function appears as time varies from 0 to 2.0 seconds. When executed, we find that $y^{(t)}$ ranges from 0 to 1.0 as t varies from 0 to 2.0, as shown in Fig. 2-8A, while the resulting plot is shown in Fig. 2-8B. The graph remains on the screen until the ENTER key is pressed, after which we are asked whether or not we wish to change the scale of either one or both axes. Fig. 2-9 shows the same plot, except now the horizontal axis varies from 0 to 0.75 seconds.

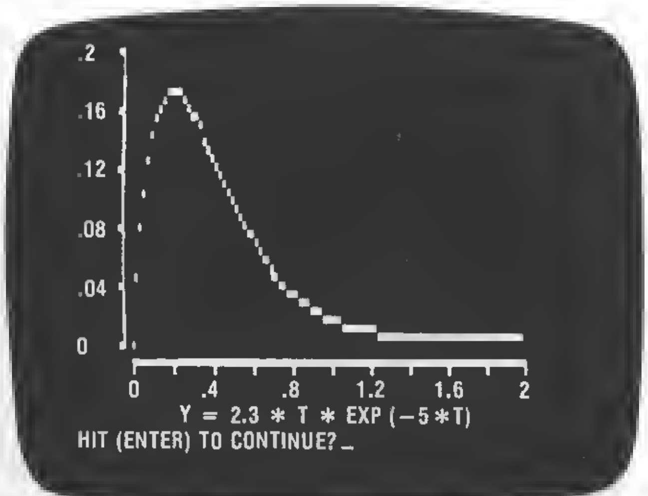
Periodic, or circular functions, such as those which have sine or cosine terms are also easily

```
100 'SAMPLE PROGRAM FOR PLOTTING A FUNCTION USING PCARTXY
101 CLS:DIM X(101),Y(101)
102 N=101:I=1
103 INPUT"MINIMUM AND MAXIMUM VALUES OF T TO BE PLOTTED ";A,B
104 C=(B-A)/100
105 FOR T=A TO B STEP C
106 Y=2.3*T*EXP(-5*T)          'SAMPLE FUNCTION
107 X(I)=T:Y(I)=Y:I=I+1
108 NEXT T
109 INPUT"TITLE FOR GRAPH ";T$
110 GOSUB 5050
111 END
```

Fig. 2-7. Main program for plotting a function of time.

```
MINIMUM AND MAXIMUM VALUES OF T TO BE PLOTTED ? 0,2
TITLE FOR GRAPH ? Y = 2.3*T*EXP(-5*T)
CURRENT MIN - MAX X VALUES ARE:  0                    2
CURRENT MIN - MAX Y VALUES ARE:  0                    .169225
DESIRED MIN,MAX X-AXIS SCALE: ? 0,2
DESIRED MIN,MAX Y-AXIS SCALE: ? 0,.2
```

(A) Output results.



(B) Video plot.

Fig. 2-8. Minimum and maximum values of T plotted.

graphed on the display, as illustrated by the following example.

Example 2-4

Plot the first three terms for the Fourier series of a square wave, such that:

$$f(x) = 1.273\sin(x) + 0.424\sin(3x) + 0.255\sin(5x)$$

Fig. 2-10A lists the main program used, plotting the function in terms of radians. Fig. 2-10B shows the output results along with the plot of this function in Fig. 2-10C.

Fig. 2-11 shows the graph for the function $y^{(t)} = 5e^{-0.2t}\sin(t)$.

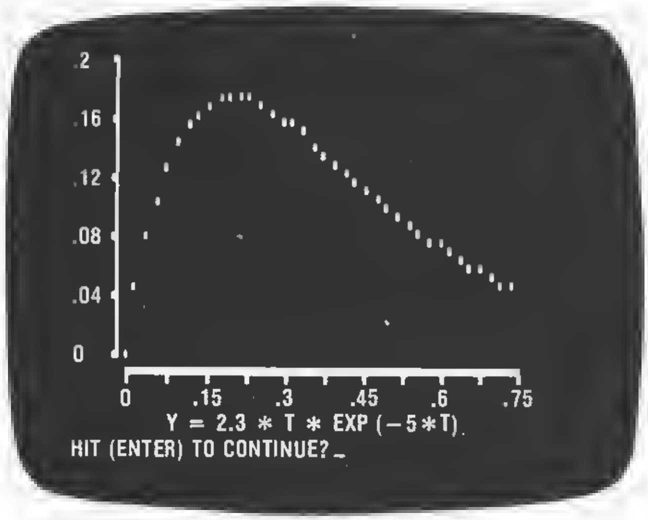


Fig. 2.9. Minimum and maximum values of T plotted with rescaled axes.

SEMILOGARITHMIC PLOTS

The PSEMILOG subroutine, listed in Fig. 2-12, permits the plotting of points on a semilogarithmic coordinate system, which is logarithmic in X and linear in Y. Depending on the range of values for the horizontal axis, the subroutine automatically scales the X-axis for values with a maximum of 4 cycles. However, the user is able to set the range for the Y-axis scale.

Example 2-5

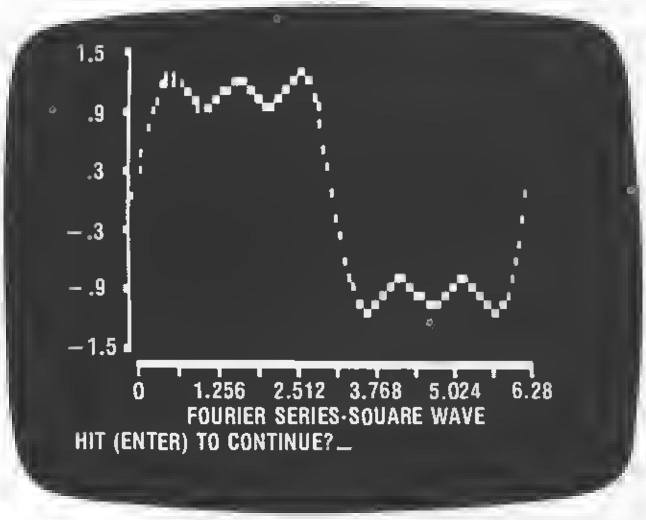
Plot the following 15 X-Y data points: (.001, 1.49), (.002,1.48), (.008,1.43), (.01,1.40), (.025, 1.29), (.05,1.15), (.088,.95), (.11,.85), (.30,.33), (.5,.12), (.94,.013), (1.23,.01), (4.3,.099), (7.1, .098), and (9.2,.098). Using the driver program of Fig. 2-13, the resulting scale limits and graph for these 15 points are shown in Fig. 2-14. The graph, which shows 4 cycles, remains on the screen until the ENTER key is pressed. Fig. 2-14B shows the plot of the same data for different axis scales.

```
100 'SAMPLE PROGRAM FOR PLOTTING A FUNCTION USING PCARTXY
101 CLS:DIM X(101),Y(101),H(6)
102 N=101:I=1
103 INPUT"ENTER MINIMUM AND MAXIUM VALUES OF X TO BE PLOTTED ";A,B
104 C=(B-A)/100
105 FOR X=A TO B STEP C
106 Y=1.273*SIN(X)+.424*SIN(3*X)+.255*SIN(5*X)     'FUNCTION TO BE PLOTTED
107 X(I)=X:Y(I)=Y:I=I+1
108 NEXT X
109 INPUT"TITLE FOR GRAPH ";T$
110 GOSUB 5050
111 END
```

(A) Main program.

```
MINIMUM AND MAXIMUM VALUES OF X TO BE PLOTTED ? 0,6.28
TITLE FOR GRAPH ? FOURIER SERIES - SQUARE WAVE

CURRENT MIN - MAX X VALUES ARE:  0                  6.28
CURRENT MIN - MAX Y VALUES ARE: -1.18677                  1.18653
DESIRED MIN,MAX X-AXIS SCALE: ? 0,6.28
DESIRED MIN,MAX Y-AXIS SCALE: ? -1.5,1.5
```

(B) Output results.



(C) Video plot.

Fig. 2-10. Plotting a function using PCARTXY.

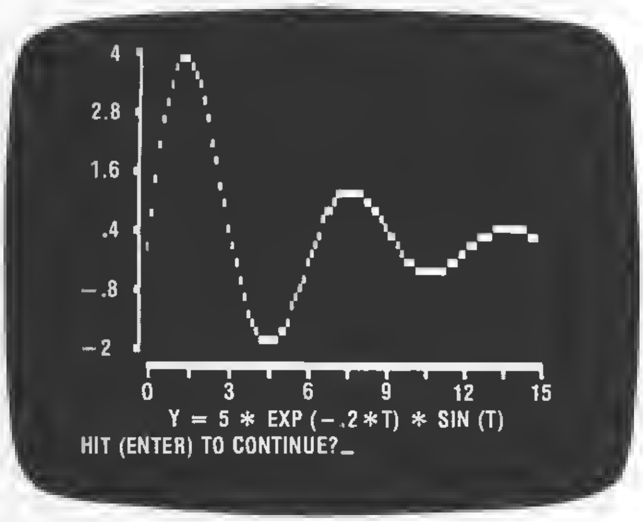


Fig. 2-11. Video plot of the function $y = 5e^{-.2t}\sin(t)$.

Continuous functions that require a semilogarithmic coordinate system are also easily graphed. These primarily include magnitude and phase plots of transfer functions (Bode plots), such as those representing active or passive filters.

Example 2-6

The transfer function,

$$T(s) = \frac{1}{s^2 + 1.414s + 1}$$

representing a simple second-order function, can then be expressed in the following forms for its magnitude and phase responses:

$$\text{Magnitude: } dB = -10\log(1 + \omega^4)$$
$$\text{Phase: } \theta = -\tan(\omega^2)$$

```
5100 ' SEMI-LOG PLOT SUBROUTINE (PSEMILOG), MAX 4 CYCLES
5101 ' X(I)=HORIZONTAL AXIS
5102 ' Y(I)=VERTICAL AXIS
5103 ' N=NUMBER OF DATA POINTS
5104 ' DETERMINE MIN & MAX VALUES OF X & Y
5105 Z9=1:Y1=Y(1):Y2=Y(1):X1=X(1):X2=X(1)
5106 FOR I=2TON
5107 IF(Y1-Y(I))<=0 THEN 5110 ELSE 5108
5108 Y1=Y(I)
5109 GOTO 5112
5110 IF(Y2-Y(I))<0 THEN 5111 ELSE 5112
5111 Y2=Y(I)
5112 IF(X1-X(I))<=0 THEN 5115 ELSE 5113
5113 X =X(I)
5114 GOTO 5117
5115 IF(X2-X(I))<0 THEN 5116 ELSE 5117
5116 X2=X(I)
5117 NEXTI
5118 PRINT"MIN - MAX X VALUES ARE: ";X1,X2
5119 PRINT"MIN - MAX Y VALUES ARE: ";Y1,Y2
5120 XY=LOG(X1)/LOG(10):XU=LOG(X2)/LOG(10):XS=X2/X1:
     AA=LOG(XS)/LOG(10):AB=INT(AA)
5121 CX=INT(ABS(XY-XU))+1
5122 IF ABS(AE-XU)-INT(ABS(AE-XU))<1E-4 THEN CX=CX-1 ELSE CX=CX
5123 AE=LOG(X1)/LOG(10):AF=INT(AE):IF(AE-AF)>.95 THEN 5124 ELSE 5125
5124 XL=10↑AE:GOTO5127
5125 XL=10↑AF
5126 AC=INT(AE):XL=10↑AC
5127 IF XL*(10↑CX)<X2 THEN CX=CX+1
5128 IF CX>4.9 THEN CLS:PRINT"DATA EXCEEDS 4 CYCLES.  CANNOT
     PLOT GRAPH":END
5129 INPUT"MIN,MAX Y-AXIS SCALE ";Y1,Y2: YD=Y2-Y1
5130 CLS
5131 PRINT@0,Y2:PRINT@128,Y1+(YD*4/5):PRINT@256,Y1+(YD*3/5)
5132 PRINT@448,Y1+(YD*2/5):PRINT@576,Y1+(YD/5):PRINT@704,Y1
5133 FOR Y=0TO35
5134 SET(15,Y)
5135 NEXTY
5136 FORX=17TO117
5137 SET(X,37)
5138 NEXTX
5139 FORY=0TO35 STEP 7
5140 SET(14,Y)
5141 NEXTY
5142 U=100/CX:FORX=17TO117 STEP U
5143 SET(X,38):SET(X,39)
5144 NEXTX
5145 Z=0
5146 FOR G=0TOCX
5147 H(G)=XL*10↑G
5148 NEXTG
5149 IF CX<>2 THEN 5152
5150 PRINT@902,H(0):PRINT@927,H(1):PRINT@952,H(2)
5151 GOTO 5164
5152 IF CX<>3 THEN 5155
5153 PRINT@902,H(0):PRINT@918,H(1):PRINT@934,H(2)
5154 PRINT@952,H(3):GOTO5167
5155 PRINT@902,H(0):PRINT@915,H(1):PRINT@927,H(2):PRINT@939,H(3)
5156 PRINT@952,H(4):GOTO5171
5157 FORI=1TON
5158 X=17.4+((LOG(X(I)/XL))*100/(LOG(10)*CX))
```

Continued on next page.

Fig. 2-12. Listing for PSEMILOG subroutine.

```
5159 Y=35-((Y(I)-Y1)*35/YD)
5160 IFX>118 OR X<16.99 OR Y<0 OR Y>35 THEN 5162 ELSE 5161
5161 SET(X,Y)
5162 NEXT I
5163 GOTO5176
5164 SET(32,38):SET(47,38):SET(56,38):SET(62,38):SET(67,38)
5165 SET(82,38):SET(97,38):SET(106,38)
5166 SET(112,38):SET(17,38):SET(117,38):GOTO 5157
5167 SET(17,38):SET(50,38):SET(83,38):SET(117,38)
5168 SET(27,38):SET(37,38):SET(43,38):SET(60,38):SET(70,38)
5169 SET(76,38):SET(80,38):SET(94,38):SET(104,38)
5170 SET(110,38):SET(114,38):SET(47,38):GOTO 5157
5171 SET(17,38):SET(42,38):SET(67,38):SET(92,38):SET(117,38)
5172 SET(24,38):SET(32,38):SET(36,38):SET(40,38)
5173 SET(49,38):SET(57,38):SET(61,38):SET(65,38)
5174 SET(74,38):SET(82,38):SET(86,38):SET(90,38)
5175 SET(99,38):SET(107,38):SET(111,38):SET(115,38):GOTO 5157
5176 INPUT"HIT <ENTER> TO CONTINUE ";FF:CLS
5177 INPUT"ANY CHANGES IN SCALE FACTORS (YES/NO) ";A$
5178 IF A$="YES" THEN 5179 ELSE 5180
5179 Z9=0:GOTO 5118
5180 RETURN
```

Fig. 2-12 (cont). Listing for PSEMILOG subroutine.

Fig. 2-15 lists the main program required, which calls the PSEMILOG subroutine. The magnitude function is defined in line 109. Line 110 transforms the variables T (i.e., ω) and Y to the arrays X(I) and Y(I), respectively, which are required by the PSEMILOG subroutine.

Fig. 2-16A shows the magnitude plot for the transfer function as frequency is chosen to vary from 0.1 to 10 Hz, or a range of two decades (cycles). Fig. 2-16B shows the phase plot for the same transfer function.

LOG-LOG PLOTS

Fig. 2-17 lists the LOGLOG subroutine used for plotting points whose range requires logarithmic scales for both axes. The main program of Fig. 2-13 can be used except that the GOSUB 5100 statement is changed to GOSUB 5200, and V(6) must be included as part of the DIMENSION statement in line 102.

Example 2-7

Graph the following 5 discrete (X,Y) points: (.00075,.457), (.003,.349), (.008,.191), (.015, .083), and (.052,.01). Using the driver program of Fig. 2-13, the resulting scale limits and graph for these 5 points are shown in Fig. 2-18 which shows that the X-axis has 3 cycles while the Y-axis has 2 cycles.

POLAR PLOTS

Polar plots are frequently used in field theory and control systems, where the plot describes both a radial distance as well as the angle (as measured counterclockwise from the horizontal) associated with it. Examples are the electric field distribution of an antenna, or Nyquist plots of the stability of feedback systems.

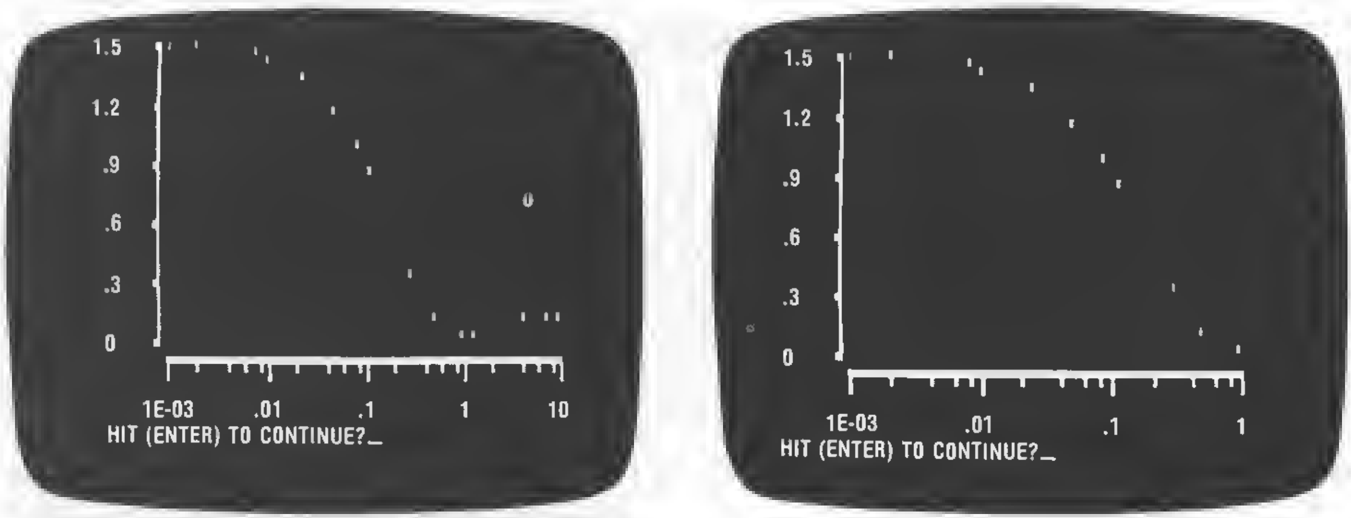
The VPOLAR program shown in Fig. 2-19 permits the plotting of either discrete data points or functions expressed in polar coordinates. The

```
100 'MAIN PROGRAM FOR PSEMILOG SUBROUTINE
101 CLS:DIM X(101),Y(101),H(6)
102 INPUT"NUMBER OF DATA POINTS TO BE PLOTTED ";N
103 FOR I=1 TO N
104 PRINT"POINT";I;
105 INPUT"X, Y ";X(I),Y(I)
106 NEXT I
107 GOSUB 5100                     'PSEMILOG SUBROUTINE
108 END
```

Fig. 2-13. Simple main program for PSEMILOG.

```
NUMBER OF DATA POINTS TO BE PLOTTED ? 15
POINT 1 X, Y ? .001,1.49
POINT 2 X, Y ? .002,1.48
POINT 3 X, Y ? .008,1.43
POINT 4 X, Y ? .01,1.40
POINT 5 X, Y ? .025,1.29
POINT 6 X, Y ? .05,1.15
POINT 7 X, Y ? .088,0.95
POINT 8 X, Y ? .11,0.85
POINT 9 X, Y ? .3,0.33
POINT 10 X, Y ? .5,0.12
POINT 11 X, Y ? .94,0.013
POINT 12 X, Y ? 1.23,0.01
POINT 13 X, Y ? 4.3,0.099
POINT 14 X, Y ? 7.1,0.098
POINT 15 X, Y ? 9.2,0.098


MIN - MAX X VALUES ARE:  1E-03              9.2
MIN - MAX Y VALUES ARE:  .01               1.49
MIN,MAX Y-AXIS SCALE: ? 0,1.5
```

(A) Output results.



(B) Video plot. *(C) Rescaled axes.*

Fig. 2-14. Number of data points plotted.

```
100 'MAIN PROGRAM USING PSEMILOG TO PLOT A FUNCTION
101 'FUNCTION IS DEFINED IN LINE 109
102 CLS:DIM X(110),Y(110)
103 I=1
104 INPUT"MIN,MAX X";A,B
105 D=LOG(B/A)/LOG(10)
106 D=INT(D+.5)
107 FOR F=0 TO D-1
108 FOR T=A*10↑F TO (B/10↑(D-F-1))+A/10↑F STEP (A*10↑F)*(D)/10
109 Y=-10*LOG(1+(T↑4))/LOG(10)       '2ND ORDER FUNCTION
110 X(I)=T:Y(I)=Y:I=I+1
111 NEXT T
112 NEXTF
113 N=I-1
114 GOSUB 116                        'PSEMILOG SUBROUTINE
115 END
```

Fig. 2-15. Simple main program for plotting a function using PSEMILOG.

(A) Magnitude plot. *(B) Phase plot.*

Fig. 2-16. Transfer function as frequency is chosen to vary from 0.1 to 10 Hz.

polar function must be defined in line 145. The variable I is in degrees and is converted to radians by dividing it by 57.29577. As examples, Table 2-2 lists the polar forms of a variety of frequently used functions, where the angle is expressed in radians.

After the data points are entered or the polar function is determined, the program asks for a graph title, and then determines the maximum radius that exists. We are then asked to enter a value for the radius which is the maximum scales for both axes. The plot is then shown on the video display, which contains markers 30 degrees apart on the outer perimeter. In addition, the total, maximum radius, X-axis (DX) and Y-axis (DY) increments are also shown.

Example 2-8

Plot the cardioid

$$r = 2(1 + \cos(\theta))$$

so that there will be a point plotted every 4 degrees. Line 145 should then read:

```
145 R(K)=2*(1+COS(I/57.29577))
```

When run the maximum radius was found to be 4.0, giving the results shown in Fig. 2-20. Other polar functions plotted using the VPOLAR routine are shown in Fig. 2-21.

PLOTTING WITH A LINE PRINTER

For those of you who have a line printer, you would probably like to make a hard-copy record of a particular graph either for inclusion in a report, or for later comparison with other results.

Table 2-2. Common Polar Functions

Name	Equation
Circle:	$r = A\cos\theta$ or $r = \lvert A\sin\theta\rvert$
Cardioid:	$r = A(1+\cos\theta)$ or $r = \lvert A(1+\sin\theta)\rvert$
Lemniscate:	$r = \lvert A\mathrm{Cos}(n\theta)\rvert$ or $r = \lvert A\sin(n\theta)\rvert$
Limacon:	$r = \lvert A + B\cos\theta\rvert$ or $r = \lvert A + B\sin\theta\rvert$
N-Leaved Rose:	$r = \lvert A\cos(n\theta)\rvert$ or $r = A\sin(n\theta)\rvert$
Archimedian Spiral:	$r = A\theta$
Hyperbolic Spiral:	$r = A/\theta$
Logarithmic Spiral:	$r = e^{B\theta}$

The following programs allow continuous functions to be graphed on virtually any line printer.

Cartesian Plots

Fig. 2-22 shows the LINEPLOT program which permits the plotting of a function, which must be defined in line 109, on a Cartesian coordinate system. The program asks for the minimum and maximum values of X for which the function is to be plotted, as well as the X-axis increment scale. In addition, the program will label both axes.

Example 2-9

Plot the function $y(t) = e^{-.2x}\sin(x)$ as x varies from 0 to 15.0 radians. Fig. 2-23 shows the resultant graph, for which the scale of the Y-axis was set to vary from −0.5 to 1.0.

```
5200 'LOG-LOG PLOT SUBROUTINE (LOGLOG)
5201 'INPUTS: X(I) AND Y(I)
5202 '        N = # OF POINTS
5203 Y1=Y(1):Y2=Y(1):X1=X(1):X2=X(1)
5204 FOR I=2TON
5205 IF(Y1-Y(I))<=0 THEN 5208 ELSE 5206
5206 Y1=Y(I)
5207 GOTO 5210
5208 IF(Y2-Y(I))<0 THEN 5209 ELSE 5210
5209 Y2=Y(I)
5210 IF(X1-X(I))<=0 THEN 5213 ELSE 5211
5211 X1=X(I)
5212 GOTO 5215
5213 IF(X2-X(I))<0 THEN 5214 ELSE 5215
5214 X2=X(I)
5215 NEXTI
5216 XS=X2/X1:AA=LOG(XS)/LOG(10):AB=INT(AA):XY=LOG(X1)/LOG(10):
     XU=LOG(X2)/LOG(10):YY=LOG(Y1)/LOG(10):YU=LOG(Y2)/LOG(10)
5217 CX=INT(ABS(XY-XU))+1
5218 IF ABS(AE-XU)-INT(ABS(AE-XU))<1E-4 THEN CX=CX-1 ELSE CX=CX
5219 AE=LOG(X1)/LOG(10):AF=INT(AE):IF(AE-AF)>.95  THEN 5220 ELSE 5221
5220 XL=10↑AE:GOTO5223
5221 XL=10↑AF
5222 AC=INT(LOG(X1)/LOG(10)):XL=10↑AC
5223 IF XL*(10↑CX)<X2 THEN CX=CX+1
5224 IF CX>4.9 THEN CLS:PRINT"X-AXIS DATA EXCEEDS 4 CYCLES.  CANNOT
      PLOT GRAPH":END
5225 YS=Y2/Y1:BA=LOG(YS)/LOG(10):BB=INT(BA)
5226 CY=INT(ABS(YY-YU))+1
5227 IF ABS(BE-YU)-INT(ABS(BE-YU))<1E-4 THEN CY=CY-1 ELSE CY=CY
5228 BE=LOG(Y1)/LOG(10):BF=INT(BE):IF(BE-BF)>.95 THEN 5229 ELSE 5230
5229 YL=10↑BE:GOTO 5232
5230 YL=10↑BF
5231 BC=INT(BE):YL=10↑BC
5232 IF YL*(10↑CY)<Y2 THEN CY=CY+1
5233 IF CY>4.9 THEN CLS:PRINT"Y-AXIS DATA EXCEEDS 4 CYCLES.  CANNOT
      PLOT GRAPH":END
5234 CLS:GOTO5261
5235 U=100/CX:FORX=17TO117 STEP U
5236 SET(X,38):SET(X,39)
5237 NEXTX
5238 Z=0
5239 FOR G=0TOCX
5240 H(G)=XL*10↑G
5241 NEXTG
5242 IF CX<>2 THEN 5245
5243 PRINT@902,H(0):PRINT@927,H(1):PRINT@952,H(2)
5244 GOTO 5250
5245 IF CX<>3 THEN 5248
5246 PRINT@902,H(0):PRINT@918,H(1):PRINT@934,H(2):PRINT@952,H(3)
5247 GOTO5252
5248 PRINT@902,H(0):PRINT@915,H(1):PRINT@927,H(2):PRINT@939,H(3):
     PRINT@952,H(4)
5249 GOTO5255
5250 SET(32,38):SET(47,38):SET(56,38):SET(62,38):SET(67,38):SET(82,38):
     SET(97,38):SET(106,38)
5251 SET(112,38):SET(17,38):SET(117,38):GOTO 5274
5252 SET(17,38):SET(50,38):SET(83,38):SET(117,38)
5253 SET(27,38):SET(37,38):SET(43,38):SET(60,38):SET(70,38):SET(76,38):
     SET(80,38):SET(94,38):SET(104,38)
5254 SET(110,38):SET(114,38):SET(47,38):GOTO 5274
```

Continued on next page.

Fig. 2-17. Listing for LOGLOG subroutine.

```
5255 SET(17,38):SET(42,38):SET(67,38):SET(92,38):SET(117,38)
5256 SET(24,38):SET(32,38):SET(36,38):SET(40,38)
5257 SET(49,38):SET(57,38):SET(61,38):SET(65,38)
5258 SET(74,38):SET(82,38):SET(86,38):SET(90,38)
5259 SET(99,38):SET(107,38):SET(111,38):SET(115,38):GOTO5274
5260 GOTO 5278
5261 FOR G=0TOCY:V(G)=YL*10↑G:NEXTG
5262 IF CY<>2 THEN 5265
5263 PRINT@0,V(2):PRINT@384,V(1):PRINT@768,V(0)
5264 GOTO 5269
5265 IF CY<>3 THEN 5268
5266 PRINT@0,V(3):PRINT@256,V(2):PRINT@512,V(1):PRINT@768,V(0)
5267 GOTO 5269
5268 PRINT@0,V(4):PRINT@192,V(3):PRINT@384,V(2):PRINT@576,V(1):
     PRINT@768,V(0)
5269 FORY=0TO36:SET(15,Y):NEXTY
5270 V=36/CY:FORY=0TO36 STEPV
5271 SET(14,Y):NEXTY
5272 FORX=17TO117:SET(X,37):NEXTX
5273 GOTO5235
5274 FOR I=1TON
5275 X=17.5+((LOG(X(I)/XL))*100/(CX*LOG(10)))
5276 Y=36.1-((LOG(Y(I)/YL))*36/(CY*LOG(10)))
5277 SET(X,Y):NEXTI
5278 INPUT"HIT <ENTER> TO CONTINUE ";Z9:CLS:RETURN
```

Fig. 2-17 (cont). Listing for LOGLOG subroutine.

Semilogarithmic Plots

The XLOGPLOT program shown in Fig. 2-24 permits the plotting of a function having a logarithmic behavior, which must be defined in line 110. Such would be the case of the frequency response of a filter, which might cover several decades.

Example 2-10

Plot the frequency response of a second-order low-pass filter, having the form:

$$dB = -10\log(1 + f^2)$$

Fig. 2-25 shows the resultant plot from 0.1 to 10 Hz.

Polar Plots

The POLARPLT program shown in Fig. 2-26 permits the plotting of polar functions (Table 2-2), such as those describing cardioids, spirals, etc., and must be defined in line 109.

When run, the program requires only a graph title and the scale limit for the two axes.

Example 2-11

Plot the N-leaved rose

$$r = \sin(3\theta)$$

As shown in Fig. 2-27, the scale limits of the two axes were arbitrarily set at 1.2.

(A) Output results.

```
NUMBER OF DATA POINTS TO BE PLOTTED ? 5
POINT 1 X, Y ? .00075,.457
POINT 2 X, Y ? .003,.349
POINT 3 X, Y ? .008,.191
POINT 4 X, Y ? .015,.083
POINT 5 X, Y ? .052,.01

MIN - MAX X VALUES ARE:  7.5E-04              .052
MIN - MAX Y VALUES ARE:  .01                  .457
```

(B) Video plot.

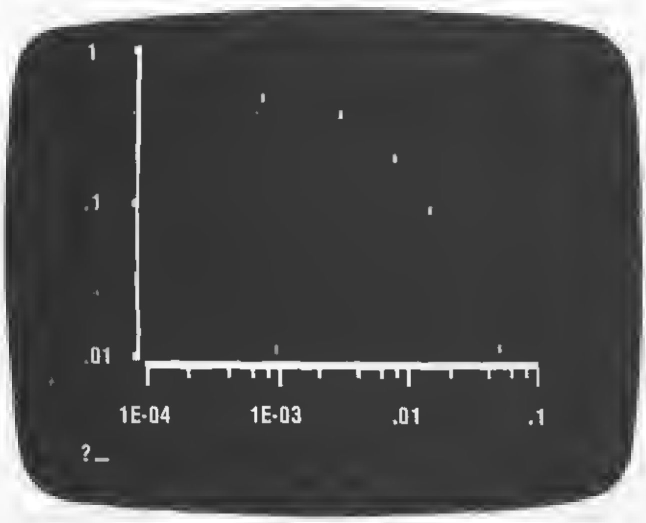


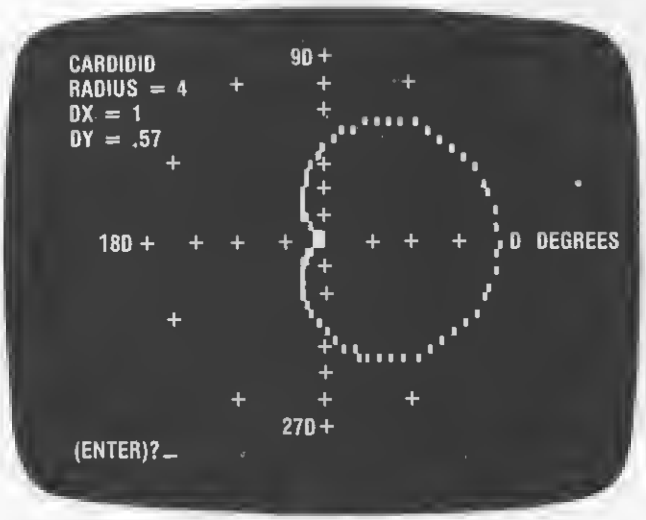
Fig. 2-18. Five data points plotted.

```
100 'POLAR PLOT - VIDEO DISPLAY (VPOLAR)
101 CLS:PRINT"GRAPH IS FOR:"
102 PRINT"      1. DISCRETE DATA POINTS
103 INPUT"      2. A POLAR FUNCTION     ";AA
104 Z9=1:PRINT:INPUT"GRAPH TITLE (MAXIMUM 20 CHARACTERS) ";T$
105 DIM Z(361),X(361),Y(361),R(361),A(361)
106 IF AA=1 THEN 107  ELSE 140
107 CLS:INPUT"NUMBER OF DATA POINTS ";N
108 FOR I=1 TO N:INPUT"MAGNITUDE, ANGLE ";R(I),Z(I)
109 A(I)=Z(I)/57.29577:X(I)=R(I)*COS(A(I)):Y(I)=R(I)*SIN(A(I))
110 NEXT I
111 Z9=2:GOTO 147
112 CLS:PRINT@0,T$:PRINT@28,"90 +"
113 PRINT@64,"RADIUS =";RM:PRINT@84,"+":PRINT@95,"+":PRINT@106,"+"
114 PRINT@128,"DX =";INT((RM/4)*100+.5)/100:PRINT@159,"+"
115 PRINT@192,"DY =";INT((RM/7)*100+.5)/100:PRINT@223,"+"
116 PRINT@268,"+":PRINT@287,"+":PRINT@306,"+":PRINT@351,"+":PRINT@415,"+"
117 PRINT@453,"180"
118 PRINT@457,"+":PRINT@463,"+":PRINT@468,"+":PRINT@474,"+":PRINT@479,"+"
119 PRINT@485,"+":PRINT@490,"+":PRINT@496,"+":PRINT@501,"+"
120 PRINT@503,"0 DEGREES":PRINT@543,"+":PRINT@607,"+":PRINT@652,"+"
121 PRINT@671,"+":PRINT@690,"+"
122 PRINT@735,"+":PRINT@799,"+":PRINT@852,"+":PRINT@863,"+":PRINT@874,"+"
123 PRINT@923,"270":PRINT@927,"+"
124 FOR K=1 TO N
125 X=63+(X(K)*44/RM):Y=22-(Y(K)*21/RM)
126 SET(X,Y):NEXT K
127 INPUT"<ENTER> ";Z$
128 CLS:PRINT"DO YOU WANT TO CHANGE:"
129 PRINT"      1. AXIS SCALES":IFZ9=2 THEN 130  ELSE 132
130 INPUT"      2. NO CHANGE      ";Z8
131 PRINT:ON Z8 GOTO 136  ,139
132 PRINT"      2. PLOT INCREMENT"
133 PRINT"      3. BOTH"
134 INPUT"      4. NO CHANGE     ";Z7
135 PRINT:ON Z7 GOTO 136  , 137  , 138  , 139
136 INPUT"MAXIMUM RADIUS ";RM:GOTO 112
137 Z9=0:GOTO 140
138 Z9=1:GOTO 140
139 END
140 K=1
141 INPUT"PLOT INCREMENT, IN DEGREES ";G
142 FOR I=0 TO 360 STEP G:CLS:PRINT"COMPUTING"
143 R(K)=25*COS(I/57.29577)      'SAMPLE FUNCTION
144 IF K>360/G THEN 147  ELSE 145
145 A(K)=I/57.29577:X(K)=R(K)*COS(A(K)):Y(K)=R(K)*SIN(A(K))
146 K=K+1:N=K:NEXT I
147 'START MIN/MAX ROUTINE
148 R1=R(1):R2=R(1)
149 FOR I=2TON
150 IF(R1-R(I))<=0 THEN 152  ELSE 151
151 R1=R(I):GOTO 154
152 IF(R2-R(I))<0 THEN 153  ELSE 154
153 R2=R(I):IF R2<1 THEN R2=1
154 NEXT I
155 IF Z9=0 THEN 112
156 CLS:PRINT"MAXIMUM RADIUS OF FUNCTION =";R2
157 INPUT"MAXIMUM RADIUS FOR AXES ";RM
158 IF RM<R2 THEN 157
159 GOTO 112
```

Fig. 2-19. Listing for VPOLAR program.

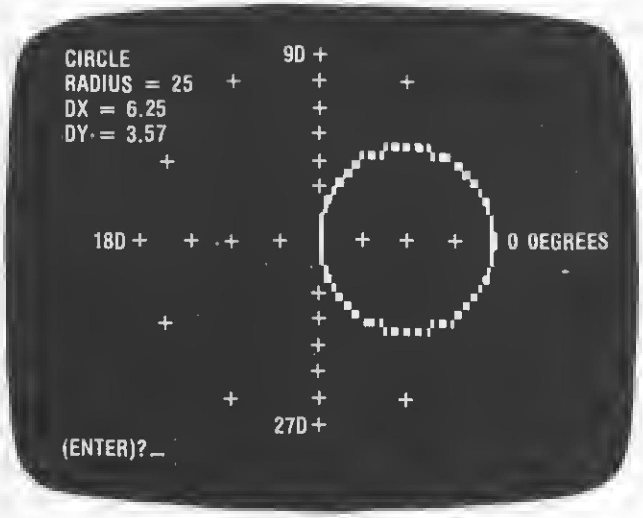
```
GRAPH IS FOR:
     1. DESCRETE DATA POINTS
     2. A POLAR FUNCTION    ? 2

ENTER:

GRAPH TITLE (MAXIMUM 20 CHARACTERS) ? CARDIOID
PLOT INCREMENT, IN DEGREES ? 4

MAXIMUM RADIUS OF FUNCTION = 4
MAXIMUM RADIUS FOR AXES ? 4
```

(A) Output results.

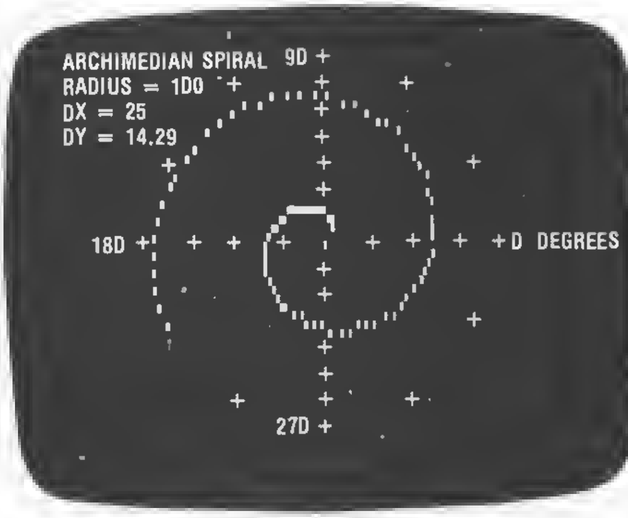


(B) Video plot.

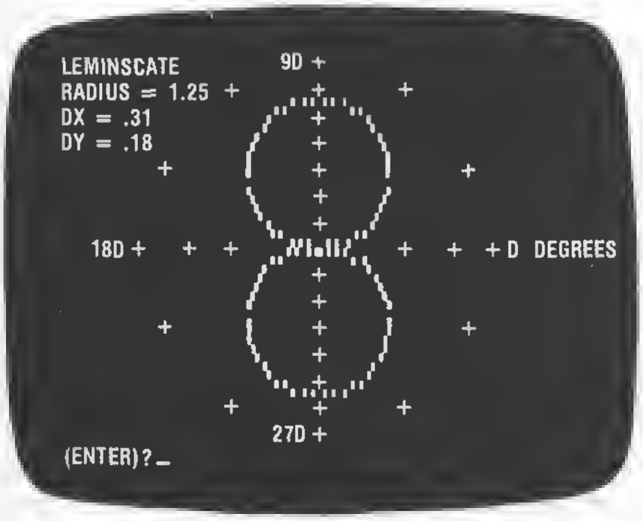
Fig. 2-20. Plotting cardioid.



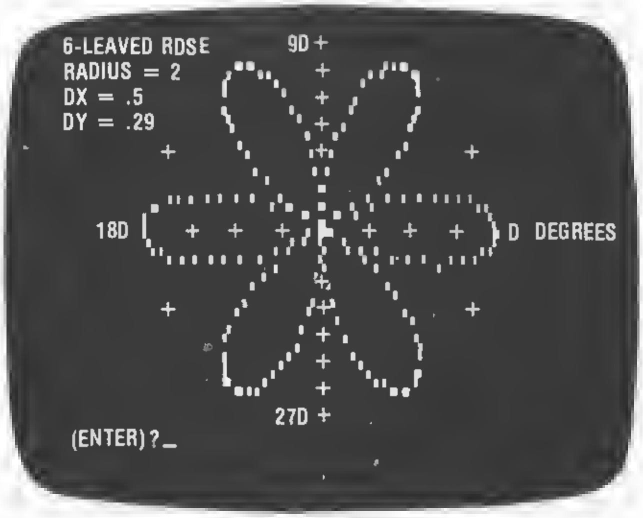
(A) Circle.



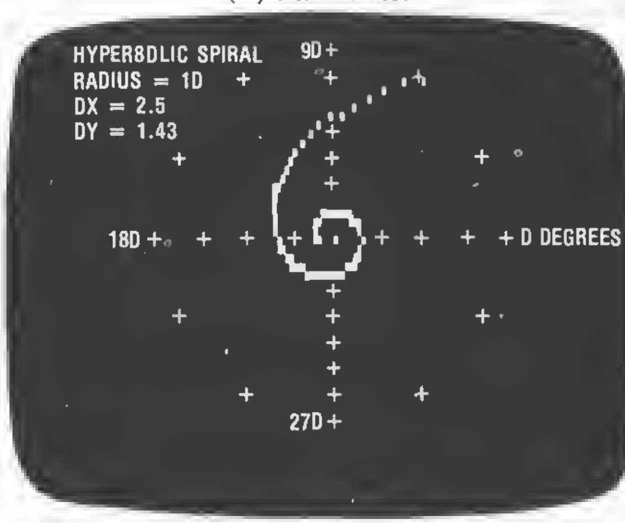
(B) Archimedian spiral.



(C) Lemniscate.



(D) 6-leaved rose.



(E) Hyperbolic spiral.

Fig. 2-21. Video plot of various polar functions.

```
100 'X-Y PLOT ROUTINE FOR LINE PRINTER (LINEPLOT)
101 CLS:DIM X(100),Y(100)
102 INPUT"MIN, MAX X VALUES OF FUNCTION TO BE PLOTTED ";X1,X2
103 INPUT"X-AXIS INCREMENT";R
104 INPUT"X-AXIS TITLE ";X$
105 K=LEN(X$)
106 INPUT"Y-AXIS TITLE ";Y$
107 I=1:XD=X2-X1:S=XD/R
108 FOR X=X1 TO X2 STEP R
109 Y=EXP(-.2*X)*SIN(X)          'SAMPLE FUNCTION
110 X(I)=X:Y(I)=Y
111 I=I+1:N=I-1
112 NEXT X
113 GOSUB 116
114 GOSUB 130
115 END
116 'DETERMINE MIN & MAX VALUES OF Y
117 Y1=Y(1):Y2=Y(1)
118 FOR I=1 TO N
119 IF(Y1-Y(I))<=0 THEN 122  ELSE 120
120 Y1=Y(I)
121 GOTO 124
122 IF(Y2-Y(I))<0 THEN 123  ELSE 124
123 Y2=Y(I)
124 NEXTI
125 PRINT"MINIMUM Y-AXIS VALUE = ";Y1
126 PRINT"MAXIMUM Y-AXIS VALUE = ";Y2
127 PRINT:INPUT"ENTER MINIMUM Y-AXIS VALUE ";Y1
128 INPUT"ENTER MAXIMUM Y-AXIS VALUE ";Y2
129 YD=Y2-Y1:RETURN
130 'PLOT POINTS
131 Z$="###.##"
132 L=YD/50
133 C1=Y1-L/2
134 D1=Y2+L/2
135 CLS
136 W=1
137 LPRINTTAB(30);Y$
138 LPRINT:FOR U=0TO5:LPRINTTAB(U*9+10+U-1);U*10*L+Y1;
139 NEXTU:LPRINT" "
140 LPRINTTAB(10);"+----+----+----+----+----+----+----+----+----+----+"
141 FOR I=1TOS+1
142 IF Y(I)<C1 THEN 149
143 IF Y(I)>=D1 THEN 149
144 LPRINT MID$(X$,W,1);:LPRINT TAB(2);:LPRINTUSINGZ$;X(I);:
145 LPRINT TAB(9)"I";TAB((Y(I)-Y1)/L+10.5);"*";TAB(61)"I"
146 W=W+1:NEXTI
147 LPRINTTAB(10);"+----+----+----+----+----+----+----+----+----+----+"
148 RETURN
149 LPRINT MID$(X$,W,1);:LPRINT TAB(2);:LPRINTUSINGZ$;X(I);:
150 LPRINT TAB(9);"I";TAB(61)"I"
151 GOTO 146
```

Fig. 2-22. Listing for LINEPLOT program.

```
MIN, MAX X VALUES OF FUNCTION TO BE PLOTTED ? 0,15.0
X-AXIS INCREMENT? 0.5
X-AXIS TITLE ? RADIANS
Y-AXIS TITLE ? AMPLITUDE

MINIMUM Y-AXIS VALUE = -.397434
MAXIMUM Y-AXIS VALUE =  .738963

ENTER MINIMUM Y-AXIS VALUE ? -0.5
ENTER MAXIMUM Y-AXIS VALUE ? 1.0
```

(*A*) *Output results.*

```
                                AMPLITUDE

          -.5       -.2        .1        .4        .7         1
           +----+----+----+----+----+----+----+----+----+----+
R    0.00 I                 *                                 I
A    0.50 I                               *                   I
D    1.00 I                                        *          I
I    1.50 I                                         *         I
A    2.00 I                                     *             I
N    2.50 I                             *                     I
S    3.00 I                   *                               I
     3.50 I           *                                       I
     4.00 I     *                                             I
     4.50 I   *                                               I
     5.00 I     *                                             I
     5.50 I         *                                         I
     6.00 I              *                                    I
     6.50 I                   *                               I
     7.00 I                      *                            I
     7.50 I                        *                          I
     8.00 I                       *                           I
     8.50 I                      *                            I
     9.00 I                   *                               I
     9.50 I                *                                  I
    10.00 I              *                                    I
    10.50 I             *                                     I
    11.00 I             *                                     I
    11.50 I              *                                    I
    12.00 I               *                                   I
    12.50 I                *                                  I
    13.00 I                  *                                I
    13.50 I                  *                                I
    14.00 I                   *                               I
    14.50 I                  *                                I
    15.00 I                  *                                I
           +----+----+----+----+----+----+----+----+----+----+
```

(*B*) *Printer plot.*

Fig. 2-23. Function $y(t) = e^{-.2x}\sin(x)$ plotted as x varies from 0 to 15 radians.

```
100 'X-Y PLOT ROUTINE FOR LINE PRINTER, SEMI-LOG IN X (XLOGPLOT)
101 CLS:DIM X(100),Y(100)
102 INPUT"MIN, MAX X";X1,X2
103 INPUT"MIN, MAX Y";Y1,Y2
104 X1=LOG(X1)/LOG(10):X2=LOG(X2)/LOG(10)
105 INPUT"X-AXIS TITLE ";X$
106 K=LEN(X$)
107 INPUT"Y-AXIS TITLE ";Y$
108 I=1:XD=X2-X1:YD=Y2-Y1:S=XD/.1
109 FOR X=X1 TO X2+.1 STEP .1
110 Y=20*LOG(1/SQR(1+((10↑X)↑2)))/LOG(10)    'SAMPLE FUNCTION
111 X(I)=X:Y(I)=Y
112 I=I+1
113 NEXT X
114 GOSUB 116
115 END
116 'PLOT GRAPH
117 Z$="####.##"
118 L=YD/50
119 C1=Y1-L/2
120 D1=Y2+L/2
121 CLS
122 W=1
123 LPRINTTAB(30);Y$
124 LPRINT:FOR U=0TO5:LPRINTTAB(U*9+10+U-1);U*10*L+Y1;
125 NEXTU:LPRINT" "
126 LPRINTTAB(10);"+----+----+----+----+----+----+----+----+----+----+"
127 FOR I=1TOS+1
128 IF Y(I)<C1 THEN 135
129 IF Y(I)>=D1 THEN 135
130 LPRINT MID$(X$,W,1);:LPRINT TAB(1);:LPRINTUSINGZ$;10↑X(I);:
131 LPRINT TAB(9)"I";TAB((Y(I)-Y1)/L+10.5);"*";TAB(61)"I"
132 W=W+1:NEXTI
133 LPRINTTAB(10);"+----+----+----+----+----+----+----+----+----+----+"
134 RETURN
135 LPRINT MID$(X$,W,1);:LPRINT TAB(1);:LPRINTUSINGZ$;10↑X(I);:
136 LPRINT TAB(9);"I";TAB(61)"I"
137 GOTO 132
```

Fig. 2-24. Listing for XLOGPLOT program.

(A) Output results.

```
MIN, MAX X? 0.1,10.0
MIN, MAX Y? -20,5
X-AXIS TITLE ? LOG FREQUENCY IN HZ
Y-AXIS TITLE ? DB RESPONSE
```

```
                        DB RESPONSE

         -20       -15       -10        -5         0         5
          +----+----+----+----+----+----+----+----+----+----+
L   0.10 I                                        *         I
O   0.13 I                                        *         I
G   0.16 I                                        *         I
    0.20 I                                        *         I
F   0.25 I                                       *          I
R   0.32 I                                       *          I
E   0.40 I                                       *          I
Q   0.50 I                                      *           I
U   0.63 I                                     *            I
E   0.79 I                                    *             I
N   1.00 I                                  *               I
C   1.26 I                                *                 I
Y   1.58 I                             *                    I
    2.00 I                          *                       I
I   2.51 I                       *                          I
N   3.16 I                   *                              I
    3.98 I               *                                  I
H   5.01 I            *                                     I
Z   6.31 I        *                                         I
    7.94 I    *                                             I
   10.00 I*                                                 I
          +----+----+----+----+----+----+----+----+----+----+
```

(B) Printer plot.

Fig. 2-25. Frequency response of low-pass filter plotted.

```
100 'POLAR PLOT FOR LINE PRINTER (POLARPLT)
101 CLEAR 200
102 DIM X(91),Y(90)
103 N=90
104 CLS:INPUT"SCALE LIMITS ";Z
105 INPUT"GRAPH TITLE ";G$
106 FOR I=1 TO N
107 CLS:PRINT"CONVERTING ANGLES TO RADIANS"
108 D=.06981317*I
109 F=2*COS(D)                    'SAMPLE FUNCTION
110 X(I)=INT(((F*COS(D)/Z+1)*30)+.5)
111 Y(I)=INT(((-F*SIN(D)/Z+1)*18)+.5)
112 NEXTI
113 CLS:PRINT"SORTING COORDINATES"
114 FOR J=1 TO N
115 FOR I=1 TO N-J
116 A=X(I):B=Y(I)
117 IF B<=Y(I+1) THEN 121
118 X(I)=X(I+1)
119 Y(I)=Y(I+1)
120 X(I+1)=A:Y(I+1)=B
121 NEXTI
122 NEXTJ
123 T=1
124 FOR P=0 TO N-1
125 IF Y(P+1)>=0 THEN 127
126 NEXTP
127 LPRINTG$
128 LPRINTTAB(29);"+Y =";Z
129 LPRINT
130 FOR I=0 TO 36
131 T=T+P
132 P=0
133 IF T>N THEN 135
134 IF Y(T)=I THEN 140
135 IF I=18 THEN 138
136 LPRINTTAB(30);"I";
137 GOTO 184
138 S=N+1
139 GOTO 172
140 FOR L=T TO N
141 IF Y(L)>Y(T) THEN 144
142 P=P+1
143 NEXTL
144 IF P=1 THEN 154
145 FOR H=1 TO P
146 FOR Q=1 TO P-H
147 C=X(T+Q-1)
148 A=X(T+Q)
149 IF C<=A THEN 152
150 X(T+Q-1)=A
151 X(T+Q)=C
152 NEXTQ
153 NEXT H
154 IF I=18 THEN 171
155 L=-1
156 S=0
157 FOR K=0 TO P-1
158 IF X(T+K)=L THEN 167
159 L=X(T+K)
160 IF L=30 THEN 164
161 IFL<30 THEN 165
```

Continued on next page.

Fig. 2-26. Listing for POLARPLT program.

```
162 IF S=1 THEN 165
163 LPRINTTAB(30);"I";
164 S=1
165 IF L>60 THEN 184
166 LPRINTTAB(L);"*";
167 NEXT K
168 IF S=1 THEN 184
169 LPRINTTAB(30);"I";
170 GOTO 184
171 S=T
172 FOR J=0 TO 60
173 IF X(S)<>J THEN 181
174 LPRINT"*";
175 FOR K=S TO T+P-1
176 IF X(K)=X(S) THEN 179
177 S=K
178 GOTO 182
179 NEXT K
180 GOTO 182
181 LPRINT"-";
182 NEXT J
183 LPRINT" +X =";Z;
184 LPRINT
185 NEXT I
186 LPRINT
187 LPRINTTAB(29);"-Y = ";Z
188 END
```

Fig. 2-26 (cont). Listing for POLARPLT program.

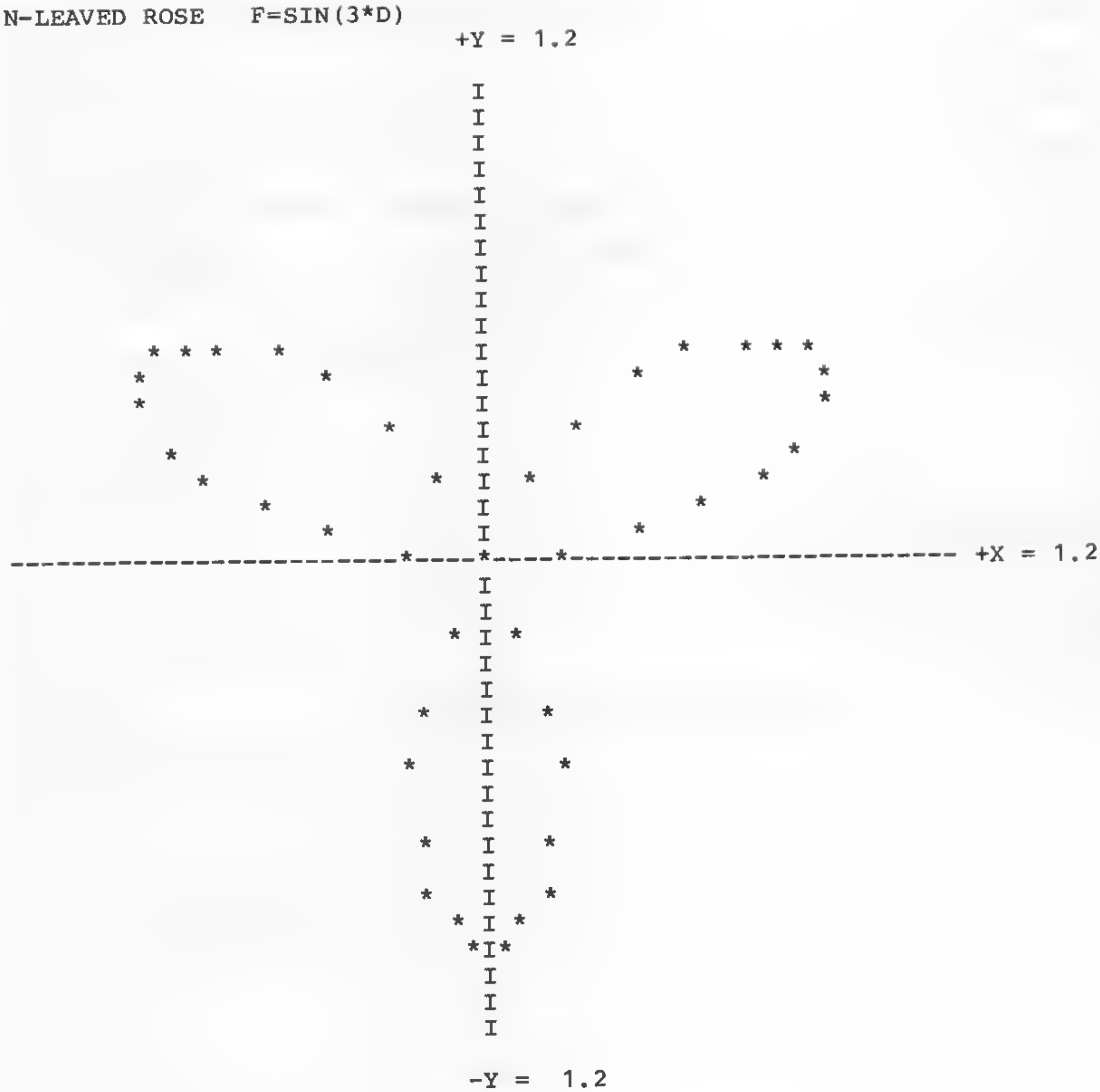


Fig. 2-27. Printer plot for Example 2-11.

3

The Analysis of Signals

This chapter on the analysis of signals discusses programs which are used to analyze both periodic and aperiodic (nonrepetitive) waveforms. In particular, we will be able to determine both the average and root mean square (rms) values of a periodic function; the representation of any periodic signal by Fourier series; the representation of aperiodic signals by a modification of the Fourier transform; and analyze damped oscillations.

```
100 'RMS & AVERAGE VALUE OF A PERIODIC WAVEFORM (RMSAV)
101 CLS:PRINT"RMS AND AVERAGE VALUE OF A PERIODIC WAVEFORM"
102 CLEAR:DIM C(361)
103 PRINT:PRINT"RMS AND AVERAGE VALUES ARE DETERMINED FROM:
104 PRINT"      1. DISCRETE POINTS
105 INPUT"      2. A DEFINED FUNCTION   ";Z9
106 PRINT:INPUT"CYCLE PERIOD ";T
107 INPUT"# DIVISIONS FOR 1 CYCLE ";C
108 V=T/C
109 IF Z9=1 THEN 110  ELSE 116
110 PRINT"TIME"
111 FOR I=0 TO C
112 PRINTI*V;
113 INPUT C(I)
114 NEXTI
115 CLS:GOTO 120
116 A=360/C                        'SAMPLE FUNCTION
117 FOR I=0 TO C                   'SAMPLE FUNCTION
118 C(I)=SIN((A*I)/57.29578)       'SAMPLE FUNCTION
119 NEXT I                         'SAMPLE FUNCTION
120 CLS:Z=1/T
121 S=0:AV=0:HC=0
122 FOR I=0 TO C-2 STEP 2
123 S=S+C(I)↑2+4*C(I+1)↑2+C(I+2)↑2
124 AV=AV+C(I)+4*C(I+1)+C(I+2)
125 IF I>C/2 THEN 126  ELSE GOSUB 137
126 PRINT@472,"COMPUTING";I
127 NEXT I
128 AV=Z*V*AV/3:IF AV<1E-3 THEN AV=0
129 HC=Z*V*HC*2/3
130 RS=SQR(Z*V*S/3)
131 PRINT"RMS VALUE =";RS
132 IF AV=0 THEN 135
133 PRINT"AVERAGE VALUE =";AV
134 IF AV<>0 THEN 136
135 PRINT"HALF CYCLE AVERAGE VALUE =";HC
136 PRINT:END
137 HC=HC+C(I)+4*C(I+1)+C(I+2):RETURN
```

Fig. 3-1. Listing for RMSAV program.

THE AVERAGE AND RMS VALUES OF A WAVEFORM

The RMSAV program listing shown in Fig. 3-1 determines both the average and rms values of a periodic signal. This is extremely useful in verifying ac measurements, such as determining the average power of a circuit, or of a particular device within the circuit. Mathematically, the average value of a periodic waveform f(t) is:

$$AV = \frac{1}{T}\int_0^T f(t)\, dt \qquad \text{(Eq. 3-1)}$$

However, it must be noted that the average value for most periodic functions having symmetry about zero amplitude (i.e., no dc term) is naturally zero, since the same amount of the waveform is above as well as below the zero reference, like most sine, triangular, and square waves. When this occurs, a half-cycle average value is determined to represent the average value of a periodic function, so that

$$AV = \frac{2}{T}\int_0^{T/2} f(t)\, dt \qquad \text{(Eq. 3-2)}$$

On the other hand, the rms value is given by:

$$RMS = \left(\frac{1}{T}\int_0^T f(t)^2\, dt\right)^{1/2} \qquad \text{(Eq. 3-3)}$$

Using Simpson's rule to numerically compute the integrals, the RMSAV program determines the average and rms values from equally spaced discrete data points, or from a periodic function, which must be supplied in line 118, so the argument of the periodic function must have the form:

argument = (A*I)/57.92577

to convert the angle from degrees to radians. If the average value is zero, then the half-cycle average is determined. Furthermore, Simpson's rule requires that the number of data points be an odd number. In addition, careful attention must be paid to discontinuities which occur during the cycle, as will be illustrated in a later example.

Example 3-1

Determine the average and rms values of an ordinary 60-Hz sine wave whose amplitude varies between −1.0 and +1.0 volts. Fig. 3-2 shows the output for the RMSAV program, when the sine wave function is supplied in line 118, giving 51 data points for one complete cycle. Since the full cycle average value for this sine wave is zero, the program then computes the half-cycle average in addition to the rms value. Using classical methods, the half-cycle average and rms values for a sine wave are 0.637 and 0.707 times the peak voltage.

Example 3-2

Consider the 50% duty cycle square wave, shown in Fig. 3-3, which varies between zero and 5 volts. We will first run the RMSAV program by entering discrete data points instead of describ-

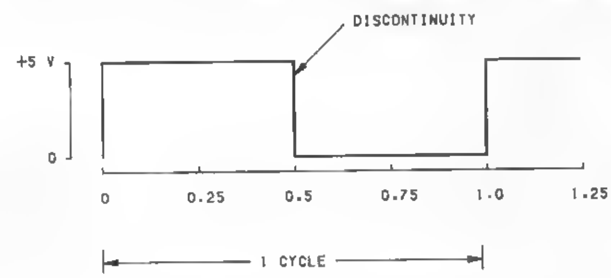


Fig. 3-3. 50% duty cycle square wave.

```
RMS AND AVERAGE VALUE OF A PERIODIC WAVEFORM

RMS AND AVERAGE VALUES ARE DETERMINED FROM:
      1. DESCRETE POINTS
      2. A DEFINED FUNCTION  ? 2

CYCLE PERIOD ? 0.16667
# DIVISIONS FOR 1 CYCLE ? 50


RMS VALUE = .707107
HALF CYCLE AVERAGE VALUE = .634111


READY
>_
```

Fig. 3-2. Output for RMSAV program.

```
RMS AND AVERAGE VALUE OF A PERIODIC WAVEFORM

RMS AND AVERAGE VALUES ARE DETERMINED FROM:
     1. DESCRETE POINTS
     2. A DEFINED FUNCTION  ? 1

CYCLE PERIOD ? 1.0
# DIVISIONS FOR 1 CYCLE ? 20

TIME
 0 ? 5
 .05 ? 5
 .1 ? 5
 .15 ? 5
 .2 ? 5
 .25 ? 5
 .3 ? 5
 .35 ? 5
 .4 ? 5
 .45 ? 5
 .5 ? 2.5
 .55 ? 0
 .6 ? 0
 .65 ? 0
 .7 ? 0
 .75 ? 0
 .8 ? 0
 .85 ? 0
 .9 ? 0
 .95 ? 0
 1 ? 0

RMS VALUE = 3.50595
AVERAGE VALUE = 2.5

READY
>_
```

Fig. 3-4. Results when 21 data points are used.

ing the function. Square waves are unusual functions in that they have discontinuities. In order to minimize computational error, it is best to assign a value that is midway between the minimum and maximum values of the square wave (i.e., 2.5 volts) as the value of the point where the discontinuity occurs during the cycle. For the situation where the square wave is discontinuous at both the beginning and the end of the cycle, we use the values of 5 and 0 volts, respectively. When numerical integration methods are used, greater accuracy is achieved when the number of points is made as large as possible. Fig. 3-4 shows the results when 21 data points are used (20 cycle divisions plus the value at $t = 0$).

Using classical methods to check our results, we would find that the average value of such a waveform is one-half its peak value, or 2.5, while the rms value is 0.707 times the peak, or 3.535. Increasing the number of data points usually leads to greater accuracy.

However, suppose we had described the square wave mathematically instead of describing it by discrete points. Unfortunately, we still have difficulty in handling the discontinuity at $t = 0.5$ second. One way around this problem is to run the program twice and then average the results. In the original RMSAV listing, lines 116–118 are replaced by the following statements:

```
116 FOR I=0 TO (C/2)-1: C(I)=5: NEXT I
117 C(C/2)=5
118 FOR I=(C/2)+1 TO C: C(I)=0: NEXT I
```

so that the results of the first run will be slightly higher than the expected value, as shown in Fig. 3-5A. Then replace line 117 by C(C/2)=0 so that the results of the second run will be slightly lower than the expected values, as shown in Fig. 3-5B. The two results are then averaged, in which case the average value is the average of 2.51667 and 2.48333, or 2.50000, while the rms value is the average of 3.5473 and 3.52373, or 3.535515.

```
RMS AND AVERAGE VALUES ARE DETERMINED FROM:
     1. DESCRETE POINTS
     2. A DEFINED FUNCTION  ? 2

CYCLE PERIOD ? 1.0
# DIVISIONS FOR 1 CYCLE ? 100

RMS VALUE = 3.5473
AVERAGE VALUE = 2.51667

READY
>_
```

(*A*) *Overestimate.*

```
RMS AND AVERAGE VALUES ARE DETERMINED FROM:
     1. DESCRETE POINTS
     2. A DEFINED FUNCTION  ? 2

CYCLE PERIOD ? 1.0
# DIVISIONS FOR 1 CYCLE ? 100

RMS VALUE = 3.52373
AVERAGE VALUE = 2.48333

READY
>_
```

(*B*) *Underestimate.*

Fig. 3-5. Rms and average value of a periodic waveform.

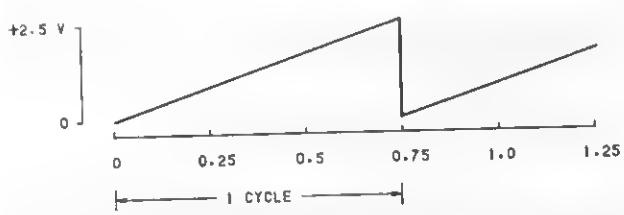


Fig. 3-6. Sawtooth waveform.

Example 3-3

Determine the average and rms values for the sawtooth waveform shown in Fig. 3-6. Unlike the square wave which has a discontinuity in the middle as well as at the beginning and end of the cycle, the sawtooth is discontinuous only at the beginning and end of the cycle.

Fig. 3-7 shows the results using 11 data points, with no corrections made for the pair of discontinuities. The starting point is zero, while the last point is the peak value of +2.5 volts. Using classical methods, the sawtooth has an average value that is one-half its peak, or 1.25 volts, while the rms value is 0.577 times the peak, or 1.443 volts.

Like the square wave, we can repeat this example by defining the waveform mathematically as a straight line with a slope equal to (2.5/0.75). Therefore, lines 116–118 would read as:

```
116 FOR I=0 TO C
117 C(I)=2.5*I/C
118 NEXT I
```

By dividing the waveform into 100 increments, the output results of Fig. 3-8 show that there is no difference!

FOURIER SERIES

Any periodic waveform can be expressed in terms of a Fourier series provided that it has a finite average value over one complete cycle; there are a finite number of discontinuities, if any exist, over one complete cycle; and it has a finite number of maximum positive and negative values. When these three conditions are met, a Fourier series for a periodic waveform f(t) exists, and can be written by the following trigonometric form:

$$f(t) = A_0 + A_1\cos(\omega t) + A_2\cos(2\omega t) + \ldots + B_1\sin(\omega t) + B_2\sin(2\omega t) + \ldots \quad \text{(Eq. 3-3)}$$

consisting of a dc, or constant term and an infinite series of harmonically related sines and cosines. However, Equation 3-3 may be stated in a form using only cosine and phase angle terms, so that:

$$f(t) = C_0 + C_1\cos(\omega t + \phi_1) + C_2\cos(2\omega t + \phi_2) + \ldots \quad \text{(Eq. 3-4)}$$

```
RMS AND AVERAGE VALUE OF A PERIODIC WAVEFORM

RMS AND AVERAGE VALUES ARE DETERMINED FROM:
     1. DESCRETE POINTS
     2. A DEFINED FUNCTION  ? 1

CYCLE PERIOD ? 0.75
# DIVISIONS FOR 1 CYCLE ? 10

TIME
 0 ? 0
 .075 ? .25
 .15 ? .5
 .225 ? .75
 .3 ? 1.0
 .375 ? 1.25
 .45 ? 1.5
 .525 ? 1.75
 .6 ? 2.0
 .675 ? 2.25
 .75 ? 2.5

RMS VALUE = 1.44338
AVERAGE VALUE = 1.25

READY
>_
```

Fig. 3-7. Results when 11 data points are used.

Equation 3-4 is sometimes referred to as the magnitude and phase Fourier series, and is related to Equation 3-3 by

$$A_n = C_n\cos(n\omega t) \qquad \text{(Eq. 3-5A)}$$

$$B_n = C_n\sin(n\omega t) \qquad \text{(Eq. 3-5B)}$$

$$C_n = (A_n^2 + B_n^2)^{1/2} \qquad \text{(Eq. 3-5C)}$$

$$\phi_n = \tan^{-1}(B_n/A_n) \qquad \text{(Eq. 3-5D)}$$

The FSERIES program listed in Fig. 3-9 calculates the A, B, and C coefficients in addition to the phase angle ϕ for a given periodic waveform. As was the case with the RMSAV program, the waveform may be expressed either as a number of equally spaced data points, or as a periodic function of time, which must be supplied in line 201, which assumes the form that 1 complete cycle equals 6.283185 radians. Line 200 sets the sampling rate as a fraction of 1 cycle. For the values shown, the function in line 201 will be computed every 1/40th of a cycle (i.e., .025), or every 9 degrees.

If equally spaced, discrete data points are used to describe the waveform, the number of data points must be an odd number, while the number of coefficients must be less than half the number of data points minus two. Like the RMSAV program, careful attention must be paid to functions that have discontinuities, whose number must be finite.

When the coefficients are evaluated, there may be some slight errors, especially since all computations are done with single precision arithmetic. However, by recognizing the symmetrical behav-

```
RMS AND AVERAGE VALUE OF A PERIODIC WAVEFORM

RMS AND AVERAGE VALUES ARE DETERMINED FROM:
     1. DESCRETE POINTS
     2. A DEFINED FUNCTION  ? 2

CYCLE PERIOD ? 0.75
# DIVISIONS FOR 1 CYCLE ? 100

RMS VALUE = 1.44338
AVERAGE VALUE = 1.25

READY
>_
```

Fig. 3-8. Results mathematically describe sawtooth wave.

```
100 'FOURIER SERIES PROGRAM
101 DIM F(100),C(100),D(100),E(100),A2(100)
102 CLS:PRINT"FOURIER SERIES TO BE COMPUTED FROM:"
103 PRINT:PRINT"     1. DISCRETE DATA POINTS"
104 INPUT"     2. AN EQUATION  ";Z9
105 IF Z9=1 THEN 106  ELSE 200
106 CLS:INPUT"NUMBER OF DATA POINTS ";N
107 IF (N/2)<>INT((N/2)+.5)  THEN 108  ELSE 106
108 INPUT"NUMBER OF HARMONICS ";N8
109 IF N8>(N/2)-2 THEN 108
110 IF Z9=1 THEN 111  ELSE 115
111 FOR I=1 TO N
112 PRINT"POINT";I;
113 INPUT F(I)
114 NEXTI
115 INPUT"SIGNAL PERIOD ";T
116 N3=N-1
117 A=2*3.1415927/T
118 T3=T/N3
119 A0=0
120 FOR I=1 TO N3 STEP 2
121 A0=A0+F(I)+4*F(I+1)+F(I+2)
122 NEXT I
123 A0=A0*T3/(3*T)
124 IF ABS(A0)<.001 THEN A0=0
125 FOR X=1 TO N8
126 CLS:PRINT@464,"CALCULATIONS IN PROGRESS"
127 PRINT@528,"HARMONIC #";X
128 C=0:D=0:E=0:A2=0
129 T1=-T3
130 FOR J=1 TO N3 STEP 2
131 T1=T1+2*T3
132 C=C+(F(J)*COS(X*A*(T1-T3)))+(4*F(J+1)*COS(X*A*T1))+
    (F(J+2)*COS(X*A*(T1+T3)))
133 D=D+(F(J)*SIN(X*A*(T1-T3)))+(4*F(J+1)*SIN(X*A*T1))+
    (F(J+2)*SIN(X*A*(T1+T3)))
134 NEXTJ
135 C=2*C*T3/(3*T):D=2*D*T3/(3*T)
136 IF ABS(C)<.005 THEN C=0
137 IF ABS(D)<.005 THEN D=0
138 E=SQR(C*C+D*D)
139 IF C>0 THEN 147  ELSE 140
140 IF C<0 AND D<0 THEN 148  ELSE 141
141 IF C=0 AND D>0 THEN 146  ELSE 142
142 IF C=0 AND D=0 THEN 143  ELSE 144
143 A2=0:GOTO 150
144 IF C=0 AND D<0 THEN 145  ELSE 149
145 A2=-90:GOTO 150
146 A2=90:GOTO 150
147 A2=(ATN(D/C))*180/3.14159:GOTO 150
148 A2=-180+(ATN(D/C))*180/3.14159:GOTO 150
149 A2=180+(ATN(D/C))*180/3.14159
150 C(X)=C:D(X)=D:E(X)=E:A2(X)=A2
151 NEXTX
152 CLS:PRINT"DC TERM, A(0) = ";A0
153 PRINT:PRINT"HARMONIC";TAB(13)"COS TERM";TAB(25)"SIN TERM";
    TAB(38)"MAGNITUDE";TAB(52)"PHASE"
154 FOR I=1TON8
155 PRINTI;TAB(13)C(I);TAB(25)D(I);TAB(38)E(I);TAB(52)A2(I):NEXTI
156 F=1/T
157 PRINT:PRINT"FUNDAMENTAL FREQUENCY =";F;"HZ"
158 PRINT:END
200 Y=1:FOR I=0 TO 1 STEP .025:     'SAMPLE FUNCTION
201 F(Y)=162*SIN(6.283185*I):       'SAMPLE FUNCTION
202 Y=Y+1:NEXT I:N=Y-1:GOTO 108  :    'SAMPLE FUNCTION
```

Fig. 3-9. Listing for FSERIES program.

ior of the particular function, one can predict which coefficients can be ignored.

Waveforms with certain types of symmetry are void of particular harmonics, such that:

1. *Odd* functions have only sine terms, so that all cosine coefficients are zero.
2. *Even* functions have only cosine terms, so that all sine terms are zero.
3. Functions that exhibit *half-wave symmetry* have only odd harmonics present.
4. Functions that are symmetrical about the independent variable axis have an average value (i.e., the dc constant term) equal to zero.

Using triangle waves, Fig. 3-10 illustrates examples of those which are odd or even functions, those having only odd harmonics, and having zero average value.

Example 3-4

Compute the Fourier series for a simple 60-Hz, 162-volt peak cosine wave. From its symmetry, we would expect that its average value is zero, and that only odd harmonic cosine terms are present. Furthermore, since we have a pure cosine wave, all harmonics except the fundamental, or first harmonic should be zero.

Lines 200 and 201 should then be stated as

```
200 Y=1:FOR I=0 TO 1 STEP .025
201 F(Y)=162*COS(6.283185*I)
```

which computes a data point every 9 degrees, or 1/40th of a cycle. Fig. 3-11 shows the results of the program. The average value is zero, while all harmonic terms are zero, except the fundamental cosine wave.

Example 3-5

Determine the Fourier series coefficients of a 10-Hz square wave whose amplitude varies between −1 and +1 volts. Since such a wave is both an odd function and has half-wave symmetry, we would then expect that the average value, all cosine terms, and all even sine terms will be zero. Fig. 3-12 shows the output results when 21 data points are used, so that:

Points 1 – 10: +1
Point 11: 0 (the half-cycle discontinuity)
Points 12 – 21: −1

As a comparison, it can be shown that the Fourier series for this square wave is:

$$f(t) = \frac{4V}{\pi}\sum_{n=0}^{m}\frac{\sin(2n+1)}{2n+1} \qquad \text{(Eq. 3-6)}$$

If desired, the square wave can be expressed mathematically as was the case for determining the average and rms values in Example 3-2, and averaging the results of two runs. This is left as an optional exercise for the reader.

Example 3-6

Determine the Fourier coefficients for the sawtooth wave previously shown in Fig. 3-6. From the symmetry of the wave, we would expect that, since it is an odd function, only sine terms will be present.

Fig. 3-13 shows the results, in which all cosine terms are zero. As a comparison, it can be shown that the Fourier series for this sawtooth is:

$$f(t) = \frac{V}{2} - \frac{V}{\pi}\sum_{n=1}^{m}\frac{\sin(n\omega t)}{n} \qquad \text{(Eq. 3-7)}$$

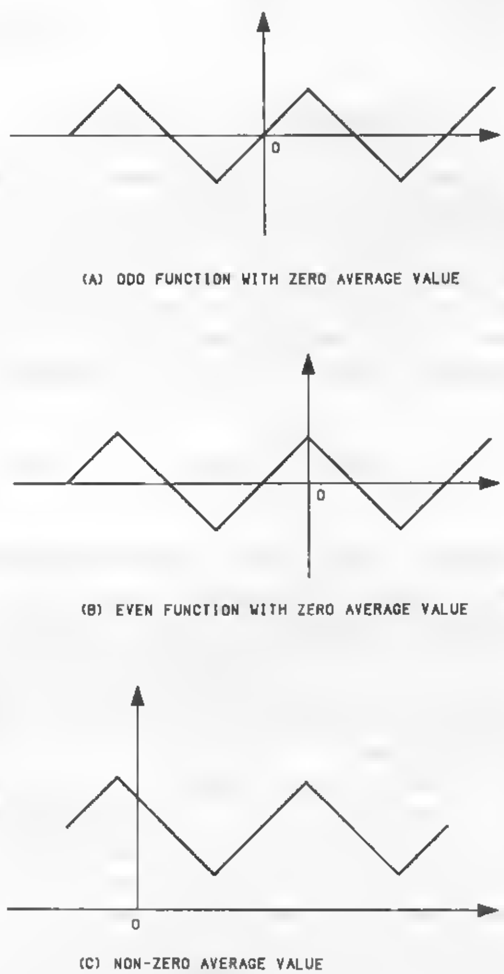


Fig. 3-10. Symmetry functions.

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DESCRETE DATA POINTS
     2. AN EQUATION  ? 2

NUMBER OF HARMONICS ? 7
SIGNAL PERIOD ? 0.01667


DC TERM, A(0) =  0

HARMONIC      COS TERM    SIN TERM     MAGNITUDE     PHASE
 1             162         0            162           0
 2             0           0            0             0
 3             0           0            0             0
 4             0           0            0             0
 5             0           0            0             0
 6             0           0            0             0
 7             0           0            0             0

FUNDAMENTAL FREQUENCY = 59.988 HZ


READY
>_
```

Fig. 3-11. Fourier series computed.

FOURIER TRANSFORM

Whereas the Fourier series is used to represent the harmonic, or spectral content of any periodic function, the Fourier transform is intended for those signals which are not periodic, but nevertheless continuous. The mathematical transformation of a function of time f(t) into a function of frequency F(s) is given by the equation:

$$F(s) = \int_{-\infty}^{\infty} f(t) e^{-st}\, dt \qquad \text{(Eq. 3-8)}$$

Instead of being able to represent the Fourier transform of a continuous function as a continuous function of frequency, the DFTRANS program of Fig. 3-14 computes a discrete Fourier transform so that the frequency function is now approximated by discrete points instead of a continuous function.

The continuous time function is sampled N times, so that $N = 2^J$, where J is an integer. If the number of sample points is 64, then J must equal 6. In the classical sense, as N is made larger, the discrete Fourier transform approaches the continuous function described by Equation 3-8.

When run, the DFTRANS program requires that the number of data point samples (N) in line 106; consequently, the value for J must also be defined, as summarized in Table 3-1. Because this program generates a spectrum plot that can contain up to 128 discrete frequencies, the maximum number of sampled data points is 256, in which case $J = 8$.

Table 3-1. DFTRANS Program

Data Points		Discrete Frequencies
N	J	
16	4	8
32	5	16
64	6	32
128	7	64
256	8	128

We have the choice to compute the Fourier transform from either by entering discrete data points describing the waveform, or mathematically defining the function. Lines 115 to 120 describe, as an example, a rectangular pulse having a duty cycle of 87.5%, so that the amplitude is logic one for 7/8ths of the period and zero for the remainder. After the input function is processed, the Fourier transform is tabulated in terms of its real and imaginary components, as well as its magnitude for each harmonic starting with the fundamental (i.e., harmonic "0"). In addition,

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DESCRETE DATA POINTS
     2. AN EQUATION  ? 1
NUMBER OF DATA POINTS ? 21
NUMBER OF HARMONICS ? 7

POINT 1 ? 1
POINT 2 ? 1
POINT 3 ? 1
POINT 4 ? 1
POINT 5 ? 1
POINT 6 ? 1
POINT 7 ? 1
POINT 8 ? 1
POINT 9 ? 1
POINT 10 ? 1
POINT 11 ? 0
POINT 12 ? -1
POINT 13 ? -1
POINT 14 ? -1
POINT 15 ? -1
POINT 16 ? -1
POINT 17 ? -1
POINT 18 ? -1
POINT 19 ? -1
POINT 20 ? -1
POINT 21 ? -1

SIGNAL PERIOD ? 0.1

DC TERM, A(0) =  0

HARMONIC     COS TERM     SIN TERM      MAGNITUDE      PHASE
 1             0            1.27331      1.27331        90
 2             0           0             0              0
 3             0           .42649        .42649         90
 4             0           0             0              0
 5             0           .266667       .266667        90
 6             0           0             0              0
 7             0           .232746       .232746        90

FUNDAMENTAL FREQUENCY = 10 HZ

READY
>_
```

Fig. 3-12. Results when 21 data points are used.

these values are displayed in groups of 10. Pressing the ENTER key continues the listing until N/2 harmonics are displayed. At the end of the listing, hitting the ENTER key once more gives the magnitude spectrum plot as a function of its harmonic number. For the program listing of Fig. 3-14, the spectrum plot will show 16 harmonics in addition to the fundamental.

Example 3-7

Compute the discrete Fourier transform for the function already described in Fig. 3-14, up to and including the 64th harmonic.

Before the program is run, line 106 must be changed so that N = 128 and J = 7. Fig. 3-15 shows the output results for the first nine harmonics plus the fundamental while the spectrum plot is shown in Fig. 3-16. Fig. 3-17 shows the spectrum plot for a 25% duty cycle pulse (64 harmonics).

ANALYSIS OF DAMPED OSCILLATIONS

When an underdamped second-order system encounters a sudden change at its input, such as

```
FOURIER SERIES TO BE COMPUTED FROM:

     1. DESCRETE DATA POINTS
     2. AN EQUATION  ? 2

NUMBER OF HARMONICS ? 7
SIGNAL PERIOD ? 0.75

DC TERM, A(0) =  1.25

HARMONIC      COS TERM     SIN TERM      MAGNITUDE     PHASE
 1              0          -.795783       .795783      -90
 2              0          -.397957       .397957      -90
 3              0          -.265499       .265499      -90
 4              0          -.199537       .199537      -90
 5              0          -.160375       .160375      -90
 6              0          -.134881       .134881      -90
 7              0          -.117562       .117562      -90

FUNDAMENTAL FREQUENCY = 1.33333 HZ

READY
>_
```

Fig. 3-13. Results with all cosine terms zero.

the voltage controlled oscillator of a phase-locked loop shifting from one frequency to another, the output tries to follow this change, but oscillates about a steady-state value for a time and eventually settles out. This process is illustrated in Fig. 3-18 for an impulse response, and in Fig. 3-19 for a step response. How fast this process is completed depends on the damping factor of the system. For smaller values of damping, it takes longer for the oscillations to settle down.

The DAMPED program listing in Fig. 3-20 determines the damping factor, the natural and damped frequencies. In addition, the program determines the normalized amplitude of the waveform at a given time. The required inputs are the positive amplitudes of two consecutive peaks, which will be one cycle apart, and the time period for one cycle. If desired, the resulting damped oscillation is plotted on the video display.

Example 3-8

From oscilloscope measurements of a damped oscillation resulting from a step response, it was measured that two consecutive peaks, like those of Fig. 3-19, had amplitudes of eight and three divisions, respectively. In addition, the period for one complete cycle was measured as 0.46 ms. We want to know what will the amplitude be after 1.5 ms, when the peak input is 0.253 volt. Fig. 3-21 shows the results of the analysis.

```
100 ' DISCRETE FOURIER TRANSFORM PROGRAM (DFTRANS)
101 CLS:PRINT,"DISCRETE FOURIER TRANSFORM PROGRAM"
102 DIM A(256), B(256), ZZ(256),X(256),Y(256)
103 PRINT:PRINT"DISCRETE FOURIER TRANSFORM IS COMPUTED FROM:"
104 PRINT"      1. DISCRETE DATA POINTS"
105 INPUT"      2. A DEFINED FUNCTION     ";Z9
106 N=32:J=5
107 IF Z9=1 THEN 108  ELSE 112
108 FOR Z=0 TO N
109 PRINTZ ;
110 INPUT"VALUE = ";A(Z)
111 NEXT Z
112 P=3.1415927
113 IF Z9=1 THEN 121
114 PRINT"CALCULATION OF INPUT FUNCTION"
115 FOR Z=0 TO N*7/8
116 A(Z)=1
117 NEXTZ
118 FOR Z=N*7/8 TO N
119 A(Z)=0
120 NEXTZ
121 FOR Z=0 TO N-1
122 A(Z)=A(Z)/N
123 NEXTZ
124 CLS:PRINT"    IN PROGRESS"
125 C=N/2:D=1:E=P*2/N
126 FOR I=1 TO J
127 F=0:G=C
128 FOR S=1 TO D
129 H=INT(F/C)
130 GOSUB 164
131 Q=R
132 A1=COS(E*Q):A2=-SIN(E*Q)
133 FOR K=F TO G-1
134 A3=A(K):A4=B(K)
135 B1=A1*A(K+C)-A2*B(K+C)
136 B2=A2*A(K+C)+A1*B(K+C)
137 A(K)=A3+B1:B(K)=A4+B2
138 A(K+C)=A3-B1:B(K+C)=A4-B2
139 NEXTK
140 F=F+2*C:G=G+2*C
141 NEXTS
142 C=C/2:D=D*2
143 NEXTI
144 GOTO 145
145 U=0
146 Z=0
147 CLS
148 PRINT"HARMONIC";TAB(10);"RE(Z)";TAB(25);"IM(Z)";TAB(39);"Z"
149 PRINT"========";TAB(10);"=====";TAB(25);"=====";TAB(37)"====="
150 H=U
151 GOSUB 172
152 PRINTTAB(2);U;TAB(8);A(R);TAB(23);B(R);TAB(37);ZZ(R)
153 Y(U)=ZZ(R):X(U)=U
154 U=U+1:Z=Z+1
155 IF Z>9 THEN 156  ELSE 158
156 PRINT:PRINT:INPUT"HIT <ENTER> TO CONTINUE ";Z9
157 GOTO 146
158 IF U>N/2 THEN 160
159 GOTO 150
160 PRINT:PRINT:INPUT"PRESS <ENTER> FOR SPECTRUM PLOT ";FF
```

Continued on next page.

Fig. 3-14. Listing for DFTRANS Program.

```
161 N=U:CLS
162 GOTO 176
163 END
164 R=0:N1=N
165 FOR W=1 TO J
166 N1=N1/2
167 IF H<N1 THEN 170
168 R=R+2↑(W-1)
169 H=H-N1
170 NEXTW
171 RETURN
172 GOSUB 164
173 ZZ(R)=SQR(A(R)↑2+B(R)↑2)
174 RETURN
175 END
176 'SPECTRUM PLOT ROUTINE
177 Y1=Y(0):Y2=Y(0):X1=X(0):X2=X(0)
178 FOR I=1 TO N
179  IF(Y1-Y(I))<=0 THEN 182  ELSE 180
180 Y1=Y(I)
181 GOTO 184
182 IF(Y2-Y(I))<0 THEN 183  ELSE 184
183 Y2=Y(I)
184 IF(X1-X(I))<=0 THEN 187  ELSE 185
185 X1=X(I)
186 GOTO 189
187 IF(X2-X(I))<0 THEN 188  ELSE 189
188 X2=X(I)+1
189 NEXTI
190 XD=X2-X1:YD=Y2-Y1
191 FOR X=0 TO 127:SET(X,36):NEXTX
192 FOR X=0 TO 127 STEP 16:SET(X,37):NEXTX:SET(127,37)
193 PRINT@831,X1:PRINT@857,"HARMONIC NUMBER":PRINT@890,X2
194 FOR I=0TO N
195 X=128-((X2-X(I))*128/XD):Y=35-((Y(I)-Y1)*35/YD)
196 IF X>127 THEN 200
197 YT=Y
198 FOR Y=YT TO 35:SET(X,Y):NEXTY
199 NEXTI
200 INPUTFF
```

Fig. 3-14 (cont). Listing for DFTRANS program.

Fig. 3-15. Output results for first nine harmonics.

HARMONIC	RE(Z)	IM(Z)	Z
========	=====	=====	=====
0	.875	0	.875
1	-.111373	-.493682	.121824
2	-.0756073	-.0834198	.112585
3	-.030777	-.0931632	.0981153
4	7.8125E-03	-.0793217	.0797055
5	.0290632	-.0513037	.0589639
6	.0302401	-.0224275	.0376491
7	.0170627	-3.83156E-03	.0174876
8	0	0	0
9	-.0111562	-7.85711E-03	.0136454

HIT <ENTER> TO CONTINUE ?

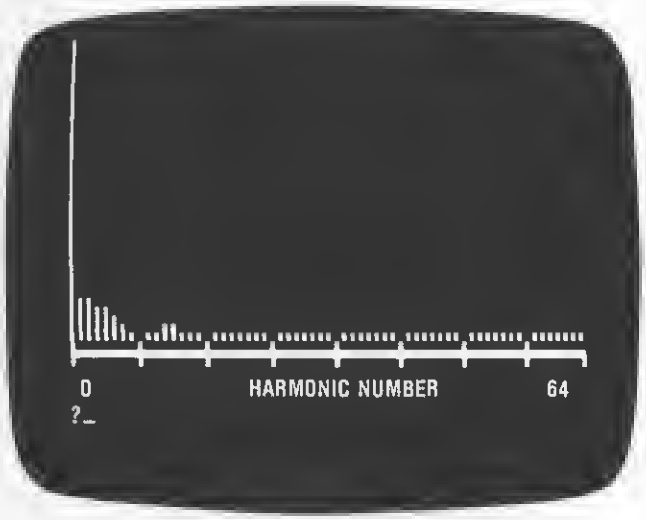


Fig. 3-16. Video spectrum plot.

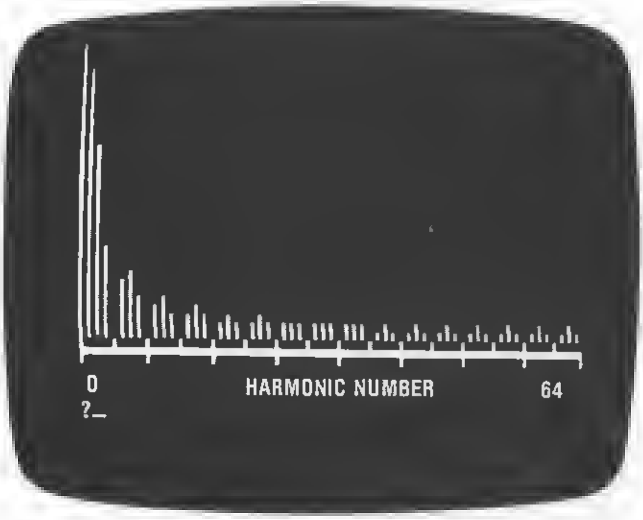


Fig. 3-17. Video spectrum plot for a 25% duty cycle pulse.

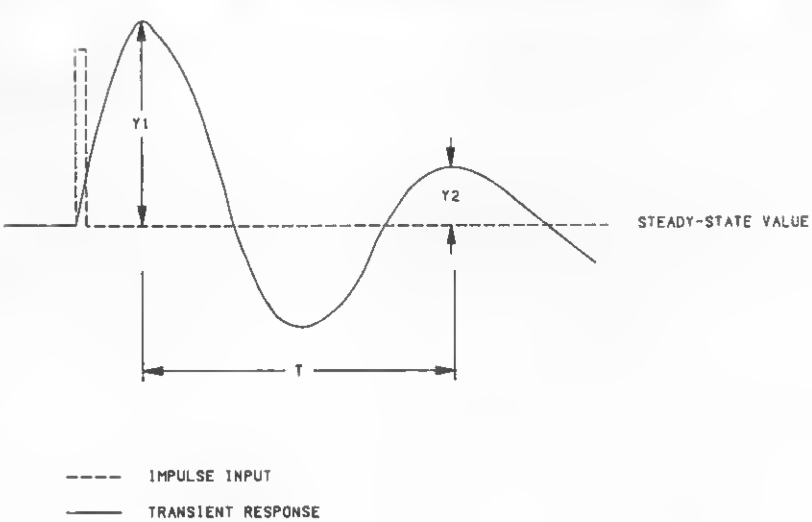


Fig. 3-18. Transient response to an impulse input.

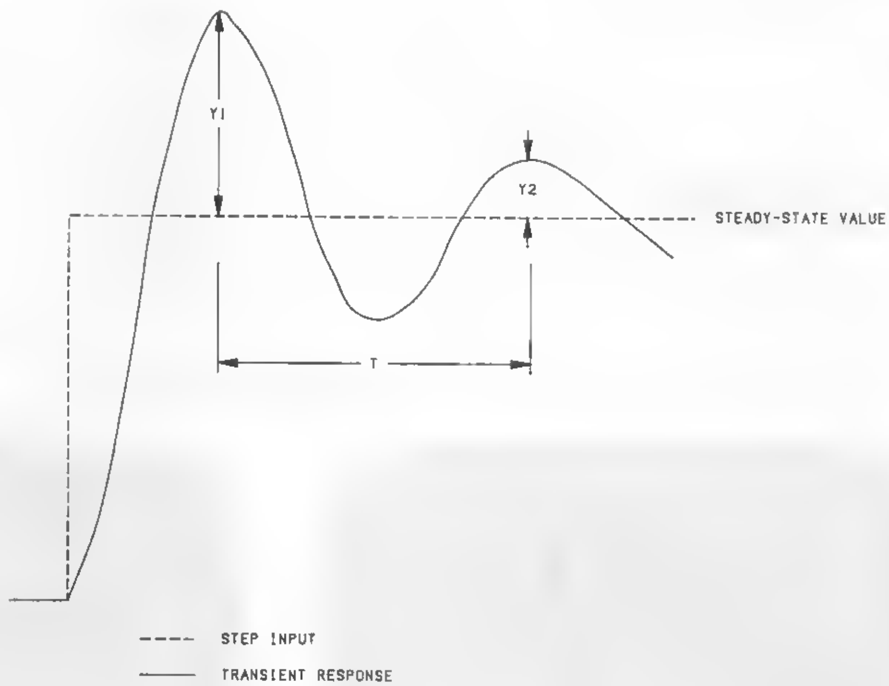


Fig. 3-19. Transient response to a step input.

```
100 'ANALYSIS OF DAMPED OSCILLATIONS (DAMPED)
101 DIM X(101),Y(101)
102 CLS:PRINT,"ANALYSIS OF DAMPED OSCILLATIONS"
103 PRINT:PRINT"AMPLITUDES ARE MEASURED FROM STEADY-STATE VALUE"
104 PRINT:INPUT"POSITIVE AMPLITUDE OF 1ST CYCLE ";Y1
105 INPUT"POSITIVE AMPLITUDE OF 2ND CYCLE ";Y2
106 INPUT"CYCLE PERIOD (IN SECONDS) ";T
107 X=Y1/Y2:G=LOG(X)*(1/6.28319)
108 D=G/SQR(1+G↑2)
109 PRINT:PRINT"THEN:"
110 G=INT(G*1000+.5)/1000
111 PRINT"DAMPING FACTOR =";G
112 FF=1/T:WN=FF*6.28319
113 FD=FF*SQR(1-D↑2):WD=FD*6.28319
114 A=D*WN:P=ATN(WN/A)
115 FF=INT(FF*10+.5)/10:WN=INT(WN*10+.5)/10
116 FD=INT(FD*10+.5)/10:WD=INT(WD*10+.5)/10
117 PRINT"NATURAL FREQUENCY =";FF;"HZ  (";WN;"RAD/SEC )"
118 PRINT"DAMPED NATURAL FREQUENCY =";FD;"HZ  (";WD;"RAD/SEC )"
119 PRINT:PRINT"DO YOU WISH TO KNOW THE RESPONSE AT A SPECIFIC TIME, OR"
120 PRINT"HAVE A GRAPH OF THE WAVEFORM"
121 INPUT"      (ENTER <T>IME, <G>RAPH, OR <N>O) ";R$
122 IF R$<>"T" AND R$<>"G" AND R$<>"N" THEN 119  ELSE 123
123 IF R$="T" OR R$="G" THEN 124  ELSE 134
124 CLS:PRINT"IS THE RESPONSE AS A RESULT OF A:"
125 PRINT@84,"";
126 INPUT"<S>TEP OR <I>MPULSE INPUT ";I$
127 IF R$="T" THEN 128  ELSE 135
128 INPUT"TIME ";T
129 IF I$="S" THEN 131  ELSE 130
130 Y=(WN/WD)*EXP(-A*T)*SIN(WD*T):GOTO 132
131 Y=1-(WN/WD)*EXP(-A*T)*SIN(WD*T+P)
132 PRINT"AMPLITUDE =";Y
133 GOTO 119
134 END
135 CLS:INPUT"MIN, MAX TIME ";XA,XB
136 N=101:I=1
137 C=(XB-XA)/100
138 FOR XT=XA TO XB STEP C
139 IF I$="S" THEN 141  ELSE 140
140 Y=(WN/WD)*EXP(-A*XT)*SIN(WD*XT):GOTO 142
141 Y=1-(WN/WD)*EXP(-A*XT)*SIN(WD*XT+P)
142 X(I)=XT:Y(I)=Y:I=I+1:NEXT XT
143 GOSUB 145
144 END
145 ' X-Y PLOT SUBROUTINE
146 IF I$="I" THEN T$="IMPULSE RESPONSE" ELSE T$="STEP RESPONSE"
147 ' DETERMINE MIN & MAX VALUES OF X & Y
148 Y1=Y(1):Y2=Y(1):X1=X(1):X2=X(1)
149 FOR I=2TON
150 IF(Y1-Y(I))<=0 THEN 153  ELSE 151
151 Y1=Y(I)
152 GOTO 155
153 IF(Y2-Y(I))<0 THEN 154  ELSE 155
154 Y2=Y(I)
155 IF(X1-X(I))<=0 THEN 158  ELSE 156
156 X1=X(I)
157 GOTO 160
158 IF(X2-X(I))<0 THEN 159  ELSE 160
159 X2=X(I)
160 NEXTI
161 PRINT"CURRENT MIN - MAX X VALUES ARE: ";X1,X2
```

Continued on next page.

Fig. 3-20. Listing for DAMPED program.

```
162 PRINT"CURRENT MIN - MAX Y VALUES ARE: ";Y1,Y2
163 INPUT"DESIRED MIN,MAX X-AXIS SCALE: ";X1,X2
164 INPUT"DESIRED MIN,MAX Y-AXIS SCALE: ";Y1,Y2
165 XD=X2-X1:YD=Y2-Y1
166 ' PLOT X-Y AXES AND SCALES
167 CLS
168 PRINT@0,Y2:PRINT@128,Y1+(YD*4/5):PRINT@256,Y1+(YD*3/5)
169 PRINT@448,Y1+(YD*2/5):PRINT@576,Y1+(YD/5):PRINT@704,Y1
170 FOR Y=0TO35
171 SET(13,Y):NEXTY
172 FOR X=15TO115:SET(X,37):NEXTX
173 FOR X=15TO115 STEP10:SET(X,38):NEXTX
174 FOR Y=0TO35 STEP 7
175 SET(12,Y)
176 NEXTY
177 PRINT@838,X1:PRINT@848,X1+(XD/5):PRINT@858,X1+(XD*2/5)
178 PRINT@868,X1+(XD*3/5):PRINT@878,X1+(XD*4/5):PRINT@888,X2
179 PRINT@916,T$
180 FOR I=1TON
181 ' PLOT POINTS
182 X=115-((X2-X(I))*100/XD):Y=35-((Y(I)-Y1)*35/YD)
183 IFX>115 THEN 185  ELSE 184
184 SET(X,Y):NEXTI
185 INPUT"HIT <ENTER> TO CONTINUE ";FF:CLS
186 INPUT"ANY CHANGES IN SCALE FACTORS (YES/NO) ";A$
187 IF A$="YES" THEN 161  ELSE 188
188 RETURN
```

Fig. 3-20 (cont). Listing for DAMPED program.

```
                ANALYSIS OF DAMPED OSCILLATIONS

AMPLITUDES ARE MEASURED FROM STEADY-STATE VALUE

POSITIVE AMPLITUDE OF 1ST CYCLE ? 8
POSITIVE AMPLITUDE OF 2ND CYCLE ? 3
CYCLE PERIOD (IN SECONDS) ? 0.46E-3

THEN:
DAMPING FACTOR = .156
NATURAL FREQUENCY = 2173.9 HZ  ( 13659.1 RAD/SEC )
DAMPED NATURAL FREQUENCY = 2147.9 HZ  ( 13495.7 RAD/SEC )

DO YOU WISH TO KNOW THE RESPONSE AT A SPECIFIC TIME, OR
HAVE A GRAPH OF THE WAVEFORM
     (ENTER <T>IME, <G>RAPH, OR <N>O) ? T
IS THE RESPONSE AS A RESULT OF A:
<S>TEP OR <I>MPULSE INPUT ? I
TIME ? 1.5E-3
AMPLITUDE = .0422664

DO YOU WISH TO KNOW THE RESPONSE AT A SPECIFIC TIME, OR
HAVE A GRAPH OF THE WAVEFORM
     (ENTER <T>IME, <G>RAPH, OR <N>O) ? G
IS THE RESPONSE AS A RESULT OF A:
<S>TEP OR <I>MPULSE INPUT ? I
MIN, MAX TIME ? 0,2E-3

CURRENT MIN - MAX X VALUES ARE:  0              1.98E-03
CURRENT MIN - MAX Y VALUES ARE: -.490691              .799866
DESIRED MIN,MAX X-AXIS SCALE: ? 0,2E-3
DESIRED MIN,MAX Y-AXIS SCALE: ? -.5,1.0
```

Fig. 3-21. Analysis of DAMPED oscillations.

4

Basic Statistics and the Analysis of Data

In any experimental endeavor, there usually comes a time when we must ask ourselves whether or not the data is a true representation of the experiment. Although this chapter is not meant to be a course in statistics, it is, nevertheless, important to include several useful statistical methods. Included are programs for computing the mean and standard deviation of a set of data, and regression analysis.

LEAST SQUARES REGRESSION CURVES

Almost every experimenter seeks to establish a relationship between two or more variables, such as the variation of impedance with frequency, or the variation of a transistor operating point as a function of ambient temperature and its current gain. As an aid in establishing a relationship (or lack of it), it is often desirable to express the relationship in the form of an equation.

Relationship Between Two Variables

Listed in Table 4-1 are several common forms of mathematical relationships used to relate the behavior of two variables.
To decide which equation should be used, it is helpful to graph the data in the form of a scatter diagram, as shown in Fig. 4-1, which can be compared against one of the general curves shown in Fig. 4-2.

Table 4-1. Common Forms of Mathematical Relationships

Linear	$y = mx + b$
Inverse	$y = 1/(mx + b)$
Nth order (polynomial)	$y = a_0 + a_1x + a_2x^2 + \ldots$
Geometric	$y = ax^b$
Exponential	$y = ae^{bx}$

Fig. 4-3 lists the REGRESSN program for determining the least squares regression of any one of the general forms listed in Table 4-1. The REGRESSN program does the following:

1. Stores and retrieves data from tape
2. Adds, deletes, or changes data
3. Determines the minimum and maximum values of data points
4. Plots a scatter diagram of entered data
5. Determines the number of data points entered
6. Computes the coefficient of determination, correlation coefficient, and standard error of estimate
7. Interpolates

Example 4-1

Determine what kind of a relationship exists between the following 10 X-Y data points, which are the parameters of voltage in millivolts, and temperature in degrees Fahrenheit for a given copper-constant thermocouple: (−2.581,−100), (−1.101,−20), (−0.674,0), (−0.404,13), (−0.150, 25), (−0.043,30), (0,31), (1.086,80), (1.542,101), and (3.711,190).

When thc REGRESSN program is run, we are first asked if we want to input data that has been previously stored on tape. If the above data has

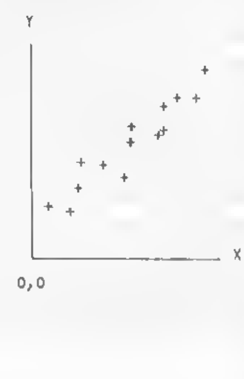


Fig. 4-1. Scatter diagram.

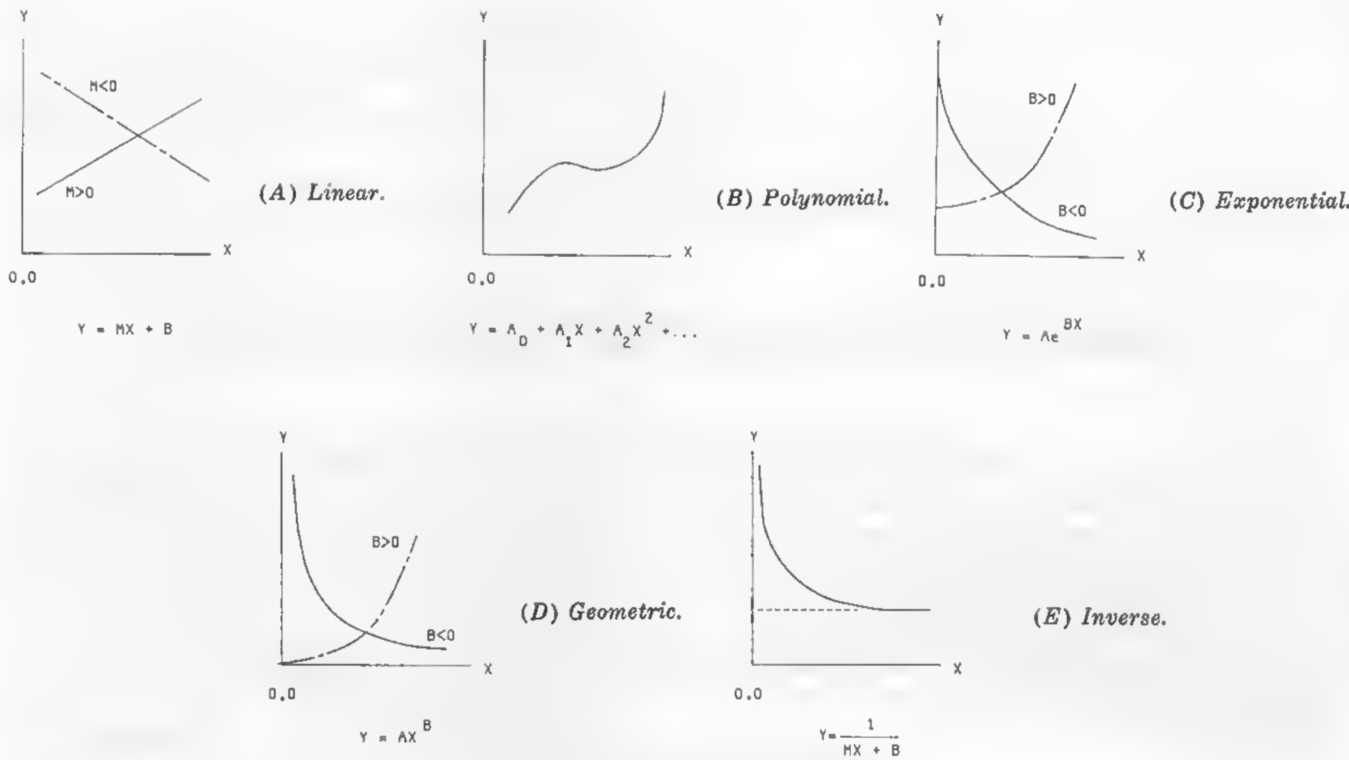


Fig. 4-2. Generalized regression curves.

not been previously stored, the answer is NO, in which case we are to enter each point, as shown in Fig. 4-4. After all data points have been entered, we must enter the point 9999,9999 to terminate. All the entered points are then listed so we may examine them for any possible errors or omissions. Suppose point No. 7 was supposed to be (0,32) instead of (0,31). We then answer YES in response to whether or not we want to change a data point. We are then asked which point is to be changed, and then the coordinates for that point.

After all corrections or additions have been made, the display then informs us of the minimum and maximum limits of our data and then gives a scatter diagram (Fig. 4-5), which can then be compared against one of the generalized curves shown in Fig. 4-2. For this example, the linear, or straight line regression appears to be the best choice. Fig. 4-6 shows the results for the linear regression line of our data, which includes the interpolation for x = −0.171. If no other interpolations are required, we can decide whether or not this data is to be saved on tape.

Because of the method in which the remaining four types of regression curves are done, the following restrictions are required:

Inverse—any x cannot equal 0.
Polynomial—order of polynomial cannot be greater than the number of data points less 2.
Geometric—all x's and y's must be greater than 0.
Exponential—any y cannot equal 0.

Although we see that it is possible to express our data in terms of a straight line, there may be another form that is better suited. For example, it might be possible that our original data may be expressed in the form of an n-th order polynomial curve. Fig. 4-7 shows the results for the polynomial regression, using a third order approximation. We see that the standard error is less, an indication that this might be the better form. When regression curves are compared, statistical tables should be consulted to determine whether or not the correlation coefficient is significant for the number of data points used.

For saving data on disk instead of tape, only a few changes are required to the basic program, as summarized below:

1. Delete lines 281 and 287.
2. Change lines 121, 141, 159, 208, 225, and 279 to 291 as shown in Fig. 4-8.

```
100 'LINEAR AND CURVILINEAR REGRESSION (REGRESSN)
101 CLEAR
102 DIM X(100),Y(100),P(100,2),V1(100),V2(10)
103 DIM V(2),R(2),M(2),D(2),A(21),U(11,12),T(12)
104 GOSUB291  :GOSUB 228  :GOSUB 246
105 GOTO 261
106 CLS:PRINT
107 CLS:PRINT"LINEAR REGRESSION"
108 GOSUB 312
109 FOR I=1 TO N:X=X(I):Y=Y(I)
110 J=J+X:K=K+Y:L=L+X[2:M=M+Y[2:R2=R2+X*Y:NEXTI
111 GOSUB268
112 PRINT
113 PRINT"Y = ";A;"+ ( ";B;"* X )"
114 GOSUB270
115 INPUT"INTERPOLATION ";C$
116 IF C$="NO" THEN 121
117 INPUT"X = ";X
118 PRINT"     THEN Y = ";A+B*X
119 INPUT"ANOTHER INTERPOLATION ";C$
120 IF C$="NO" THEN 121  ELSE 117
121 INPUT"SAVE INPUT DATA ON TAPE ";ZZ$
122 IF ZZ$="YES" THEN GOSUB 279  ELSE 267
123 GOTO 267
124 'GEOMETRIC REGRESSION ROUTINE
125 GOSUB 312
126 FOR I=1 TO N:X=X(I):Y=Y(I):Y=LOG(Y):X=LOG(X)
127 J=J+X:K=K+Y:L=L+X[2:M=M+Y[2:R2=R2+X*Y
128 NEXTI
129 GOSUB 268
130 CLS:PRINT"GEOMETRIC REGRESSION (FOR X,Y > 0)":PRINT
131 PRINT"      B":PRINT"Y = AX"
132 PRINT:PRINT"     A =";EXP(A)
133 PRINT"     B =";B
134 GOSUB 270
135 INPUT"INTERPOLATION ";C$
136 IF C$="NO" THEN 141
137 INPUT"X = ";X
138 PRINT"     THEN Y = ";EXP(A)*(X[B)
139 INPUT"ANOTHER INTERPOLATION ";C$:IF C$="NO" THEN 141
140 GOTO 137
141 INPUT"SAVE INPUT DATA ON TAPE ";ZZ$
142 IF ZZ$="YES" THEN GOSUB279  ELSE 261
143 'EXPONENTIAL REGRESSION ROUTINE
144 GOSUB 312
145 FOR I=1 TO N:X=X(I):Y=Y(I):Y=LOG(Y)
146 J=J+X:K=K+Y:L=L+X[2:M=M+Y[2:R2=R2+X*Y:NEXTI
147 GOSUB 268
148 CLS:PRINT"EXPONENTIAL REGRESSION (FOR Y<>0)":PRINT
149 PRINT"         BX"
150 PRINT"Y = A(EXP)"
151 PRINT:PRINT"     A =";EXP(A):PRINT"     B =";B
152 GOSUB 270
153 INPUT"INTERPOLATION ";C$
154 IF C$="NO" THEN 159
155 INPUT"X = ";X
156 PRINT"     THEN Y = ";EXP(A)*EXP(B*X)
157 INPUT"ANOTHER INTERPOLATION ";C$:IF C$="NO" THEN 159
158 GOTO 155
159 INPUT"SAVE INPUT DATA ON TAPE ";ZZ$
160 IF ZZ$="YES" THEN GOSUB 279  ELSE 161
161 GOTO 267
```

Continued on next page.

Fig. 4-3. Listing for REGRESSN program.

```
162 CLS:PRINT"POLYNOMIAL REGRESSION"
163 PRINT"                           N"
164 PRINT"Y = A  + A X +....A X"
165 PRINT"    0    1         N"
166 PRINT:INPUT"ORDER OF POLYNOMIAL ";D
167 CLS:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT"CALCULATIONS IN PROGRESS"
168 A(1)=N:J=0:K=0:Q=0:S=0:Z=0:I=0:CC=0
169 FOR I=1TON
170 X=X(I):Y=Y(I)
171 FORJ=2TO2*D+1:A(J)=A(J)+X[(J-1):NEXTJ
172 FORK=1TOD+1:U(K,D+2)=T(K)+Y*X[(K-1):T(K)=T(K)+Y*X[(K-1):NEXTK
173 T(D+2)=T(D+2)+Y[2:NEXTI
174 FORJ=1TOD+1
175 FORK=1TOD+1:U(J,K)=A(J+K-1):NEXTK
176 NEXTJ
177 FORJ=1TOD+1
178 FORK=JTOD+1
179 IF U(K,J)<>0 THEN 182  ELSE 180
180 NEXTK
181 PRINT:PRINT"NO UNIQUE SOLUTION CAN BE FOUND":GOTO210
182 FORI=1TOD+2:S=U(J,I):U(J,I)=U(K,I):U(K,I)=S:NEXTI
183 Z=1/U(J,J)
184 FORI=1TOD+2:U(J,I)=Z*U(J,I):NEXTI
185 FORK=1TOD+1
186 IFK=J THEN 189  ELSE 187
187 Z=-U(K,J)
188 FORI=1TOD+2:U(K,I)=U(K,I)+Z*U(J,I):NEXTI
189 NEXTK
190 NEXTJ
191 CLS:PRINT"POLYNOMIAL COEFFICIENTS:":PRINT"A( 0 ) = ";U(1,D+2)
192 FORJ=1TOD:PRINT"A(";J;") = ";U(J+1,D+2):NEXTJ
193 P=0:FORJ=2TOD+1:P=P+U(J,D+2)*(T(J)-A(J)*T(1)/N):NEXTJ
194 Q=T(D+2)-T(1)[2/N:Z=Q-P:I=N-D-1:L=P/Q
195 PRINT:INPUT"HIT ENTER TO CONTINUE ";ZZ
196 PRINT"                            2"
197 PRINT"COEFFICIENT OF DETERMINATON R  =";L
198 PRINT"CORRELATION COEFFICIENT R =";SQR(L)
199 PRINT"STANDARD ERROR OF ESTIMATE =";SQR(Z/I)
200 INPUT"INTERPOLATION ";C$
201 IF C$="NO" THEN 208
202 INPUT"X = ";X
203 P=U(1,D+2)
204 FOR J=1TOD:P=P+U(J+1,D+2)*X[J:NEXTJ
205 PRINT"     THEN Y = ";P
206 INPUT"ANOTHER INTERPOLATION ";C$
207 IFC$="YES" THEN 202
208 INPUT"SAVE INPUT DATA ON TAPE ";ZZ$
209 IF ZZ$="YES" THEN GOSUB 279  ELSE 211
210 FOR I=1TO 500:NEXT
211 GOTO 267
212 CLS:PRINT"INVERSE REGRESSION"
213 GOSUB 312
214 FOR I=1 TO N:X=1/X(I):Y=Y(I)
215 J=J+X:K=K+Y:L=L+X[2:M=M+Y[2:R2=R2+X*Y:NEXTI
216 GOSUB 268
217 PRINT:PRINT"Y = ";A;" + (";B;"/X)"
218 GOSUB 270
219 INPUT"INTERPOLATION ";C$
220 IF C$="NO" THEN 225
221 INPUT"X = ";X
222 PRINT"     THEN Y = ";A+B/X
```

Fig. 4-3 (cont) Listing

```
223 INPUT"ANOTHER INTERPOLATION ";C$
224 IFC$="NO" THEN 225  ELSE 221
225 INPUT"SAVE INPUT DATA ON TAPE ";ZZ$
226 IF ZZ$="YES" THEN GOSUB 279  ELSE 227
227 GOTO 267
228 Y1=Y(1):Y2=Y(1):X1=X(1):X2=X(1)
229 FORI=2TON
230 IF(Y1-Y(I))<=0 THEN 233  ELSE 231
231 Y1=Y(I)
232 GOTO235
233 IF(Y2-Y(I))<0 THEN 234  ELSE 235
234 Y2=Y(I)
235 IF(X1-X(I))<=0 THEN 238  ELSE 236
236 X1=X(I)
237 GOTO240
238 IF(X2-X(I))<0 THEN 239  ELSE 240
239 X2=X(I)
240 NEXTI
241 CLS:PRINT"YOUR INPUT DATA HAS THE FOLLOWING LIMITS:"
242 PRINT:PRINT:PRINT:PRINT"VARIABLE","MINIMUM","MAXIMUM"
243 PRINT"X",X1,X2:PRINT"Y",Y1,Y2
244 XD=X2-X1:YD=Y2-Y1
245 RETURN
246 PRINT:INPUT"HIT <ENTER> FOR GRAPH OF INPUT DATA ";FF
247 CLS:PRINT@0,Y2:PRINT@128,Y1+(YD*4/5):PRINT@256,Y1+(YD*3/5)
248 PRINT@448,Y1+(YD*2/5):PRINT@576,Y1+(YD/5):PRINT@704,Y1
249 FORY=0TO35:SET(13,Y):NEXTY
250 FORX=15TO115:SET(X,37):NEXTX
251 FORX=15TO115STEP20:SET(X,38):NEXTX
252 FORY=0TO35STEP7:SET(12,Y):NEXTY
253 PRINT@838,X1:PRINT@848,X1+(XD/5):PRINT@858,X1+(XD*2/5)
254 PRINT@868,X1+(XD*3/5):PRINT@878,X1+(XD*4/5):PRINT@888,X2
255 PRINT@916,"INPUT DATA - Y VS. X"
256 FORI=1TON
257 X=115-((X2-X(I))*100/XD):Y=35-((Y(I)-Y1)*35/YD)
258 SET(X,Y):NEXTI
259 INPUT"HIT <ENTER> TO CONTINUE ";FF
260 RETURN
261 CLS:PRINT"REGRESSION TYPES AVAILABLE:"
262 PRINT:PRINT"1. LINEAR REGRESSION":PRINT"2. INVERSE REGRESSION"
263 PRINT"3. NTH ORDER (POLYNOMIAL) REGRESSION"
264 PRINT"4. GEOMETRIC REGRESSION":PRINT"5. EXPONENTIAL REGRESSION"
265 PRINT:INPUT"CHOICE ";ZX
266 ON ZX GOTO 107  ,212  ,162  ,124  ,143
267 END
268 B=(N*R2-K*J)/(N*L-J[2)
269 A=(K-B*J)/N:RETURN
270 J=B*(R2-J*K/N)
271 M=M-K[2/N
272 K=M-J:IF K<0 THEN K=0
273 R2=J/M:PRINT:PRINT"NUMBER OF DATA POINTS =";N
274 PRINT"                                2"
275 PRINT"COEFFICIENT OF DETERMINATION R  =";R2
276 PRINT"CORRELATION COEFFICIENT R =";SQR(R2)
277 PRINT"STANDARD ERROR OF ESTIMATE =";SQR(K/(N-2))
278 RETURN
279 'STORE DATA POINTS ON TAPE
280 PRINT"LOAD TAPE AND PRESS RECORD AND PLAY BUTTONS"
281 INPUT"PRESS <ENTER> WHEN READY ";Z9
282 PRINT#-1,N
283 FOR I=1 TO N:PRINT#-1,X(I),Y(I):NEXT I
284 RETURN
```

Continued on next page.

for REGRESSN program.

```
285 'INPUT DATA POINTS FROM TAPE
286 PRINT"LOAD TAPE AND PRESS PLAY BUTTON"
287 INPUT"PRESS <ENTER> WHEN READY ";Z9
288 INPUT#-1,N
289 FOR I=1 TO N:INPUT#-1,X(I),Y(I):NEXT I
290 RETURN
291 CLS:INPUT"INPUT PREVIOUSLY STORED DATA FROM TAPE (YES/NO) ";ZZ$
292 IF ZZ$="YES" THEN GOSUB 285  ELSE 294
293 GOSUB300  :IF ZZ$="YES" THEN RETURN
294 CLS:PRINT"ENTER EACH DATA POINT (ENTER 9999,9999 TO TERMINATE): "
295 N=1
296 INPUT"     X,Y ";X(N),Y(N)
297 IF (X(N)=9999)AND(Y(N)=9999) THEN 299  ELSE 298
298 N=N+1:GOTO 296
299 N=N-1
300 CLS:PRINT"POINT","X","Y"
301 FOR VV=1 TO N:PRINTVV,X(VV),Y(VV):NEXTVV
302 PRINT:INPUT"CHANGE ANY DATA POINT(S) ";C$
303 IF C$="YES" THEN 304  ELSE 307
304 GOSUB 314
305 C$="":INPUT"ANOTHER CHANGE ";C$
306 IF C$="YES" THEN 304  ELSE 307
307 INPUT"DELETE A DATA POINT ";D$
308 IF D$="YES" THEN  318  ELSE 309
309 INPUT"DO YOU WANT TO ADD OTHER DATA POINTS ";C$
310 IF C$="YES" THEN 311  ELSE RETURN
311 N=N+1:GOTO 296
312 I=0:J=0:K=0:L=0:M=0:Q=0:S=0:Z=0
313 CC=0:R2=0:RETURN
314 CLS:PRINT"POINT","X","Y"
315 FOR VV=1 TO N:PRINTVV,X(VV),Y(VV):NEXT VV
316 PRINT:INPUT"CHANGE POINT # ";CC
317 INPUT"NEW X,Y ";X(CC),Y(CC):RETURN
318 CLS:PRINT"POINT","X","Y"
319 FOR VV=1 TO N:PRINTVV,X(VV),Y(VV):NEXT VV
320 PRINT:INPUT"DELETE POINT # ";KK
321 FOR L=KK TO N:X(L)=X(L+1):Y(L)=Y(L+1):NEXTL
322 N=N-1:GOTO 300
```

Fig. 4-3 (cont). Listing for REGRESSN program.

```
INPUT PREVIOUSLY STORED DATA FROM TAPE (YES/NO) ? NO
ENTER EACH DATA POINT (ENTER 9999,9999 TO TERMINATE):
     X,Y ? -2.581,-100
     X,Y ? -1.101,-20
     X,Y ? -.674,0
     X,Y ? -.404,13
     X,Y ? -.15,25
     X,Y ? -.043,30
     X,Y ? 0,31
     X,Y ? 1.086,80
     X,Y ? 1.542,101
     X,Y ? 3.711,190
     X,Y ? 9999,9999
POINT          X              Y
 1          -2.581          -100
 2          -1.101          -20
 3          -.674            0
 4          -.404            13
 5          -.15             25
 6          -.043            30
 7           0               31
 8           1.086           80
 9           1.542           101
 10          3.711           190
CHANGE DATA POINT(S) ? YES
POINT          X              Y
 1          -2.581          -100
 2          -1.101          -20
 3          -.674            0
 4          -.404            13
 5          -.15             25
 6          -.043            30
 7           0               31
 8           1.086           80
 9           1.542           101
 10          3.711           190
CHANGE POINT # ? 7
NEW X,Y ? 0,32
ANOTHER CHANGE ? NO
DELETE A DATA POINT ? NO
DO YOU WANT TO ADD OTHER DATA POINTS ? NO
```

Fig. 4-4. Input data for Example 4-1.

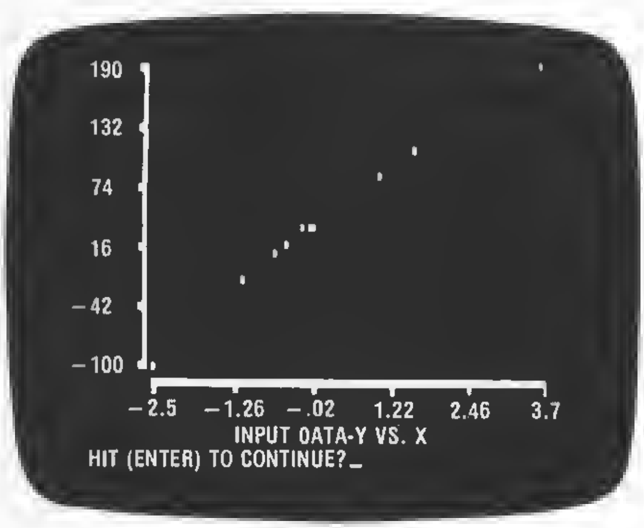


Fig. 4-5. Video scatter diagram of Example 4-1.

```
LINEAR REGRESSION
Y = 28.7844 + ( 45.5669 * X )

NUMBER OF DATA POINTS = 10
                                       2
COEFFICIENT OF DETERMINATION R  = .995586
CORRELATION COEFFICIENT R = .997791
STANDARD ERROR OF ESTIMATE = 5.43301
INTERPOLATION ? YES
X = ? -.171
     THEN Y =  20.9925
ANOTHER INTERPOLATION ? NO
SAVE INPUT DATA ON TAPE ? YES

READY
>_
```

Fig. 4-6. Linear regression output results.

```
POLYNOMIAL REGRESSION
                                          N
Y = A   + A X +....A X
     0     1         N

ORDER OF POLYNOMIAL ? 3

POLYNOMIAL COEFFICIENTS:
A( 0 ) =  32.0932
A( 1 ) =  46.1803
A( 2 ) = -1.52834
A( 3 ) =  .149678

HIT ENTER TO CONTINUE ?
                                            2
COEFFICIENT OF DETERMINATON R  = .999966
CORRELATION COEFFICIENT R = .999983
STANDARD ERROR OF ESTIMATE = .553163
INTERPOLATION ? YES
X = ? -.171
     THEN Y =  24.151
ANOTHER INTERPOLATION ? NO
SAVE INPUT DATA ON TAPE ? NO

READY
>_
```

Fig. 4-7. Third order polynomial regression output results.

Consequently, a data file with the file name DATA will be created on the same disk.

Example 4-2

Table 4-2 summarizes the data points from an experiment that measured the thermal resistance of a heat sink as a function of its volume. From past experience we might know that this should look like a straight line when plotted using log-log coordinates, thus suggesting a geometric regression line.

Fig. 4-9 shows the output for the data of Table 4-2 using a geometric regression.

```
GEOMETRIC REGRESSION (FOR X,Y > 0)
       B
Y = AX

     A = 46.155
     B =-1.52145

NUMBER OF DATA POINTS = 9
                                  2
COEFFICIENT OF DETERMINATION R  = .999152
CORRELATION COEFFICIENT R = .999576
STANDARD ERROR OF ESTIMATE = .0793054
INTERPOLATION ? YES
X = ? 4.0
     THEN Y =  5.60032
ANOTHER INTERPOLATION ? NO

SAVE INPUT DATA ON TAPE ? NO

READY
>_
```

Fig. 4-9. Geometric regression results.

Relationship Between More Than Two Variables

The REGRESSN program dealt only with the possible relationships that may exist between a

Table 4-2. Data for Example 4-2

Thermal Resistance (degrees/watt)	Volume (in^3)
12.0	1.0
10.0	1.3
8.0	2.0
4.0	6.0
1.3	30.0
0.8	70.0
0.4	200
0.3	300
0.13	900

```
121 INPUT"SAVE INPUT DATA ON DISK ";ZZ$
141 INPUT"SAVE INPUT DATA ON DISK ";ZZ$
159 INPUT"SAVE INPUT DATA ON DISK ";ZZ$
208 INPUT"SAVE INPUT DATA ON DISK ";ZZ$
225 INPUT"SAVE INPUT DATA ON DISK ";ZZ$

279 'STORE DATA POINTS ON DISK
280 OPEN"O",1,"DATA"
282 PRINT#1,N
283 FOR I=1 TO N:PRINT#1,X(I),Y(I):NEXT I
284 CLOSE:RETURN
285 'INPUT DATA POINTS FROM DISK
286 OPEN"I",1,"DATA"
288 INPUT#1,N
289 FOR I=1 TO N:INPUT#1,X(I),Y(I):NEXT I
290 CLOSE:RETURN
291 CLS:INPUT"INPUT PREVIOUSLY STORED DATA FROM DISK (YES/NO) ";ZZ$
```

Fig. 4-8. Required changes to REGRESSN program for disk operation.

```
100 'MULTIPLE LINEAR REGRESSION PROGRAM (MLREGRES)
101 CLS:PRINT,"MULTIPLE LINEAR REGRESSION"
102 DIM X(30),S(30),T(30),A(30,31)
103 PRINT:INPUT"NUMBER OF DATA POINTS ";N
104 INPUT"NUMBER OF INDEPENDENT VARIABLES ";V
105 CLS
106 PRINT"ENTER DEPENDENT AND INDEPENDENT VARIABLES FOR EACH DATA POINT"
107 PRINT:X(1)=1
108 FOR I=1 TO N:PRINT:PRINT"POINT";I
109 INPUT"          DEPENDENT VARIABLE Y ";X(V+2)
110 FOR J=1 TO V
111 PRINT"          INDEPENDENT VARIABLE A(";J;") ";
112 INPUT X(J+1)
113 NEXT J
114 FOR K=1 TO V+1
115 FOR L=1 TO V+2
116 A(K,L)=A(K,L)+X(K)*X(L)
117 S(K)=A(K,V+2)
118 NEXT L
119 NEXT K
120 S(V+2)=S(V+2)+X(V+2)↑2
121 NEXTI
122 FOR I=2 TO V+1
123 T(I)=A(1,I)
124 NEXT I
125 FOR I=1 TO V+1
126 FOR J=I TO V+1
127  IF A(J,I)<>0 THEN 131
128 NEXT J
129 PRINT"NO UNIQUE SOLUTION"
130 GOTO 176
131 FOR K=1 TO V+2
132 B=A(I,K)
133 A(I,K)=A(J,K)
134 A(J,K)=B
135 NEXTK
136 Z=1/A(I,I)
137 FOR K=1 TO V+2
138 A(I,K)=Z*A(I,K)
139 NEXT K
140 FOR J=1 TO V+1
141 IF J=I THEN 146
142 Z=-A(J,I)
143 FOR K=1 TO V+2
144 A(J,K)=A(J,K)+Z*A(I,K)
145 NEXT K
146 NEXT J
147 NEXT I
148 CLS:PRINT"RESULTS OF MULTIPLE LINEAR REGRESSION":PRINT
149 PRINT"A( 0 ) = ";A(1,V+2)
150 FOR I=2 TO V+1
151 PRINT"A(";I-1;") = ";A(I,V+2)
152 NEXT I
153 P=0
154 FOR I=2 TO V+1
155 P=P+A(I,V+2)*(S(I)-T(I)*S(1)/N)
156 NEXT I
157 R=S(V+2)-S(1)↑2/N
158 Z=R-P
159 L=N-V-1
160 I=P/V
161 I=P/R
```

Continued on next page.

Fig. 4-10. Listing for MLREGRES program.

```
162 PRINT:PRINT"                              2"
163 PRINT"COEFFICIENT OF DETERMINATION R  = ";I
164 PRINT"CORRELATION COEFFICIENT R =";SQR(I)
165 PRINT"STANDARD ERROR OF ESTIMATE = ";SQR(ABS(Z/L))
166 PRINT:INPUT"INTERPOLATION ";C$
167 IF C$="YES" THEN 168  ELSE 176
168 P=A(1,V+2)
169 FOR J=1 TO V
170 PRINT"INDEPENDENT VARIABLE A(";J;") ";
171 INPUTX
172 P=P+A(J+1,V+2)*X:NEXT J
173 PRINT"     THEN Y = ";P
174 INPUT"ANOTHER INTERPOLATION ";C$
175 IF C$="YES" THEN 168
176 PRINT:END
```

Fig. 4-10 (cont). Listing for MLREGRES program.

dependent variable and a single independent variable. However, there are situations where the measured quantity is dependent upon two or more quantities, which may be interrelated or may not be interrelated.

The multiple linear regression program (MLREGRES) listed in Fig. 4-10 determines the coefficients for the form:

$$y = a_0 + a_1x_1 + a_2x_2 \ldots + a_Nx_N$$

In addition, the program determines the coefficients of determination and correlation, as well as the standard error of estimate.

Example 4-3

For the following data, determine the multiple regression coefficients, if a solution can be found from the given data.

dependent variable y	64	71	53	67	55	58
independent variable x_1	57	59	49	62	51	50
independent variable x_2	8	10	6	11	8	7

As shown in the output results of Fig. 4-11, the dependent variable is entered first for each point, followed by the two independent variables, x_1 and x_2. The regression coefficients then yield the resulting equation:

$$y = -5.03233 + 1.21878x_1 - 0.0313288x_2$$

In addition, the interpolation is shown for the values $x_1 = 52$ and $x_2 = 10$.

SAMPLE STATISTICS

The DSTATS program listed in Fig. 4-12 determines the arithmetic mean, variance, standard deviation, skewness, kurtosis, z-scores, maximum and minimum values, and the range for a sample of ungrouped data. The variance and standard deviation are measures of dispersion of the data about the mean value. The coefficients of skewness and kurtosis are measures of asymmetry and peakness, respectively, of a distribution of data. The z-scores are measures of the deviation from the mean in terms of a standard deviation, so that other distributions of data may be compared.

When run the DSTATS program displays the value, its difference from the mean, and Z-score in groups of 10. Hitting the ENTER key continues the rest of the list. After the entire list of values are displayed, the remaining statistics of the sample are given.

Example 4-4

Determine the sample statistics for the following 15 measured values for 100-ohm resistors: 99, 103, 99, 98, 100, 101, 105, 97, 99, 100, 102, 101, 103, 98, and 101.

As shown in Fig. 4-13, the mean value for these 15 resistors over the range of 97 to 105 is 100.4 ohms, while the variance and standard deviation are 4.83036 and 2.19781 ohms, respectively.

```
                    MULTIPLE LINEAR REGRESSION

NUMBER OF DATA POINTS ? 6
NUMBER OF INDEPENDENT VARIABLES ? 2
ENTER DEPENDENT AND INDEPENDENT VARIABLES FOR EACH DATA POINT


POINT 1
          DEPENDENT VARIABLE Y ? 64
          INDEPENDENT VARIABLE A( 1 ) ? 57
          INDEPENDENT VARIABLE A( 2 ) ? 8

POINT 2
          DEPENDENT VARIABLE Y ? 71
          INDEPENDENT VARIABLE A( 1 ) ? 59
          INDEPENDENT VARIABLE A( 2 ) ? 10

POINT 3
          DEPENDENT VARIABLE Y ? 53
          INDEPENDENT VARIABLE A( 1 ) ? 49
          INDEPENDENT VARIABLE A( 2 ) ? 6

POINT 4
          DEPENDENT VARIABLE Y ? 67
          INDEPENDENT VARIABLE A( 1 ) ? 62
          INDEPENDENT VARIABLE A( 2 ) ? 11

POINT 5
          DEPENDENT VARIABLE Y ? 55
          INDEPENDENT VARIABLE A( 1 ) ? 51
          INDEPENDENT VARIABLE A( 2 ) ? 8

POINT 6
          DEPENDENT VARIABLE Y ? 58
          INDEPENDENT VARIABLE A( 1 ) ? 50
          INDEPENDENT VARIABLE A( 2 ) ? 7


RESULTS OF MULTIPLE LINEAR REGRESSION

A( 0 ) = -5.03233
A( 1 ) =  1.21878
A( 2 ) = -.0313288

                                2
COEFFICIENT OF DETERMINATION R   =  .838228
CORRELATION COEFFICIENT R = .915548
STANDARD ERROR OF ESTIMATE =  3.69589

INTERPOLATION ? YES
INDEPENDENT VARIABLE A( 1 ) ? 52
INDEPENDENT VARIABLE A( 2 ) ? 10
     THEN Y =  58.031
ANOTHER INTERPOLATION ? NO


READY
>_
```

Fig. 4-11. Multiple linear regression results.

```
100 'DESCRIPTIVE STATISTICS (DSTATS)
101 CLEAR:DIM N(100):CLS
102 INPUT"NUMBER OF OBSERVATIONS";N
103 M=0:B=0:C=0:T=0:U=0
104 FOR I=1TON:PRINT"#";I;
105 INPUTX:N(I)=X
106 C=C+X:B=B+X↑2:NEXTI
107 M=C/N:V=(B-N*M↑2)/(N-1):T=V*(N-1)/N:U=SQR(T)
108 D=SQR(V)
109 INPUT"HIT <ENTER> FOR LISTING OF VALUES AND Z-SCORES ";Z9
110 P=0:Q=0:R=0:I=1:S=1
111 CLS:PRINT" #","VALUE","DIFFERENCE","Z-SCORE"
112 P=N(I)-M:Z=P/D
113 Q=Q+(P↑3):R=R+(P↑4)
114 PRINT I,N(I),P,Z
115 IF I=N THEN 119  ELSE 116
116 IF I>(10*S)-1 THEN 118  ELSE 117
117 I=I+1:GOTO 112
118 S=S+1:PRINT:INPUT"HIT <ENTER> TO CONTINUE LIST ";Z9:I=I+1:GOTO 111
119 PRINT:PRINT"END OF LIST":PRINT
120 INPUT"HIT <ENTER> FOR SAMPLE STATISTICS ";Z9
121 N1=N(1):N2=N(1)
122 FOR I=2 TO N
123 IF(N1-N(I))<=0 THEN 125  ELSE 124
124 N1=N(I):GOTO 127
125 IF(N2-N(I))<0 THEN 126  ELSE 127
126 N2=N(I)
127 NEXT I
128 CLS:PRINT"MINIMUM VALUE = ";N1:PRINT"MAXIMUM VALUE = ";N2:
    PRINT"RANGE =";ABS(N2-N1)
129 PRINT"MEAN =";M:PRINT"VARIANCE =";V:PRINT"STD. DEVIATION =";D
130 PRINT"SKEWNESS = ";Q/(N*(U[3))
131 PRINT"KURTOSIS = ";R/(N*(U[4))
132 PRINT:END
```

Fig. 4-12. Listing for DSTATS program.

Fig. 4-13. Results of sample statistics for 15 resistor values.

```
NUMBER OF OBSERVATIONS? 15
# 1 ? 99
# 2 ? 103
# 3 ? 99
# 4 ? 98
# 5 ? 100
# 6 ? 101
# 7 ? 105
# 8 ? 97
# 9 ? 99
# 10 ? 100
# 11 ? 102
# 12 ? 101
# 13 ? 103
# 14 ? 98
# 15 ? 101
HIT <ENTER> FOR LISTING OF VALUES AND Z-SCORES ?

 #              VALUE           DIFFERENCE      Z-SCORE
 1               99             -1.4            -.636999
 2               103             2.6             1.183
 3               99             -1.4            -.636999
 4               98             -2.4            -1.092
 5               100            -.400002        -.182
 6               101             .599999         .272999
 7               105             4.6             2.02299
 8               97             -3.4            -1.547
 9               99             -1.4            -.636999
 10              100            -.400002        -.182

HIT <ENTER> TO CONTINUE LIST ?

 #              VALUE           DIFFERENCE      Z-SCORE
 11              102             1.6             .727998
 12              101             .599999         .272999
 13              103             2.6             1.183
 14              98             -2.4            -1.092
 15              101             .599999         .272999

END OF LIST

HIT <ENTER> FOR SAMPLE STATISTICS ?

MINIMUM VALUE =  97
MAXIMUM VALUE =  105
RANGE = 8
MEAN = 100.4
VARIANCE = 4.83036
STD. DEVIATION = 2.19781
SKEWNESS =  .431235
KURTOSIS =  2.4851

READY
>_
```

5

Networks and Transforms

This chapter presents programs for the design of a variety of attenuators and lossless pads, which are primarily used for impedance matching. In addition, programs are included which permit the analysis of two-port networks, the inverse Laplace transform, real and imaginary roots of polynomials, Pi-Tee (Delta-Wye) transformations, and the solution of mesh and node equations.

ROOTS OF POLYNOMIALS

Very often the analysis of a network requires that the resultant polynomial be factored into distinct roots, which may be either real or complex. Such may be the case of solving a high order differential equation by Laplace transforms, or determining the pole locations of Butterworth and Chebyshev polynomials.

The PLYROOTS program shown in Fig. 5-1 determines the real or complex roots, if any exist, for any polynomial of the form:

$$P(x) = a_0 + a_1x + a_2x^2 \ldots + a_Nx^N \quad \text{(Eq. 5-1)}$$

Naturally, the number of roots will be equal to the degree of the polynomial. It should be noted that third- and higher-order polynomials frequently have roots that are conjugate pairs in which the real parts are equal but the imaginary parts are of opposite sign. The coefficients of the polynomial are entered one at a time, starting with the constant term on up to the nth-order term.

Example 5-1

Determine the roots, if any exist, for the second-order polynomial

$$P(x) = x^2 + x - 6$$

As shown in Fig. 5-2 the coefficients of the polynomial are entered one at a time, starting with the constant term on up to the second-order term. For this example, the roots were found to be equal to 2 and −3, both of which are real. This solution agrees with the solution obtained using the quadratic formula method for second-order equations.

Example 5-2

Determine the pole locations for the following fourth-order Butterworth polynomial:

$$P(s) = 1 + 2.613s + 3.414s^2 + 2.613s^3 + s^4$$

Shown in Fig. 5-3 are the output results. We see that all four roots are complex, and exist as two complex conjugate pairs. From these roots, we know that the pole locations lie 22.5 degrees apart on the perimeter of a semicircle of unit length in the left half of the complex plane, as shown in Fig. 5-4.

THE INVERSE LAPLACE TRANSFORM

The inverse Laplace transform

$$f(t) = \frac{1}{j2\pi}\int_{\sigma_1 - j\infty}^{\sigma_1 + j\infty} F(s)e^{st}\,ds \quad \text{(Eq. 5-2)}$$

is frequently used for the solution of linear differential equations by transforming the problem from the frequency domain to the time varying response.

The inverse Laplace transform program (ILAPLACE) shown in Fig. 5-5 assumes that the frequency domain, or transfer function of the complex variable s ($s = j\omega$), is expressed by the following generalized form:

$$F(s) = \frac{a_0 + a_1s + a_2s^2 + \ldots}{(s + b_1)(s + b_2)\ldots} \quad \text{(Eq. 5-3)}$$

The polynomial coefficients a_i of the numerator are assumed to be real, while the denominator has

```
100 'REAL AND COMPLEX ROOTS OF A REAL POLYNOMIAL (PLYROOTS)
101 DIM A(40),C(40),RR(20),RI(20)
102 CLS:PRINT,"REAL & COMPLEX ROOTS OF A REAL POLYNOMIAL"
103 PRINT:INPUT"ORDER OF POLYNOMIAL ";O
104 PRINT:PRINT"ENTER COEFFICIENTS, STARTING WITH THE CONSTANT TERM":PRINT
105 FOR I=1 TO O+1
106 PRINT"COEFFICIENT A(";I-1;") ";
107 INPUT A(I)
108 NEXT I
109 IT=0:N=O
110 IF A(N+1)<>0 THEN 111  ELSE 112
111 IF N>0 THEN 112  ELSE 195
112 NX=O
113 NY=N+1
114 N2=1
115 KJ=N+1
116 FOR L=1 TO KJ
117 MT=KJ-L+1
118 C(MT)=A(L)
119 NEXT L
120 'SET INITIAL VALUES
121 X0=.00500101:Y0=.01000101
122 IN=0
123 X=X0
124 X0=-10*Y0:Y0=-10*X
125 'SET X AND Y TO CURRENT VALUE
126 X=X0:Y=Y0
127 IN=IN+1:GOTO 132
128 IT=1
129 XP=X
130 YP=Y
131 'EVALUATE POLYNOMIAL & DERIVATIVES
132 IC=0
133 UX=0:UY=0:V=0:YT=0:XT=1.00
134 U=C(N+1)
135 IF U <>0 THEN 136  ELSE 176
136 FOR I=1 TO N
137 L=N-I+1
138 T=C(L)
139 X2=X*XT-Y*YT:Y2=X*YT+Y*XT
140 U=U+T*X2
141 V=V+T*Y2
142 F1=I
143 UX=UX+F1*XT*T
144 UY=UY-F1*YT*T
145 XT=X2:YT=Y2:NEXTI
146 SQ=(UX↑2)+(UY↑2)
147 IF SQ<> 0 THEN 148  ELSE 168
148 DX=(V*UY-U*UX)/SQ
149 X=X+DX
150 DY=-(U*UY+V*UX)/SQ
151 Y=Y+DY
152 IF ABS(DY)+ABS(DX)-1E-5 <0 THEN 159  ELSE 154
153 'STEP INTERATION COUNTER
154 IC=IC+1
```

Fig. 5-1. Listing for

```
155 CLS:PRINT"INTERATION COUNT =";IC
156 IF IC-500<0 THEN 133  ELSE 157
157 IF IT<>0 THEN 159  ELSE 158
158 IF IN-5 <0 THEN 123  ELSE 195
159 FOR L=1TO NY
160 MT=KJ-L+1
161 T=A(MT)
162 A(MT)=C(L)
163 C(L)=T:NEXTL
164 IP=N
165 N=NX
166 NX=IP
167 IF IT<>0 THEN 170  ELSE 128
168 IF IT <>0 THEN 169  ELSE 123
169 X=XP:Y=YP
170 IT=0
171 IF ABS(Y)-(1E-4*ABS(X)) <0 THEN 179  ELSE 172
172 AL=X+X
173 SQ=(X↑2)+Y↑2
174 N=N-2
175 GOTO 183
176 X=0
177 NX=NX-1
178 NY=NY-1
179 Y=0
180 SQ=0
181 AL=X
182 N=N-1
183 C(2)=C(2)+AL*C(1)
184 FOR L=2 TO N
185 C(L+1)=C(L+1)+AL*C(L)-SQ*C(L-1)
186 NEXT L
187 RI(N2)=Y:RR(N2)=X
188 N2=N2+1
189 IF SQ<>0 THEN 190  ELSE 193
190 Y=-Y
191 SQ=0
192 GOTO 187
193 IF N>0 THEN 121  ELSE 194
194 GOTO 197
195 CLS:PRINT"UNABLE TO COMPUTE AFTER 500 INTERATIONS"
196 END
197 'OUTPUT RESULTS
198 CLS:PRINT"FOR THE POLYNOMIAL EQUATION:"
199 PRINT:PRINT"0 = ";A(1);" + ";A(2);"X ";
200 FOR K=3 TO O+1
201 PRINT" + ";A(K);"X↑";K-1;
202 NEXTK
203 PRINT:PRINT:PRINT"THE ROOTS ARE:"
204 PRINT:PRINT"ROOT #","REAL PART","IMAGINARY PART"
205 FOR I=1TO O
206 PRINT I,RR(I),RI(I):NEXTI
207 PRINT:PRINT
208 END
```

PLYROOTS program.

```
                REAL & COMPLEX ROOTS OF A REAL POLYNOMIAL

ORDER OF POLYNOMIAL ? 2

ENTER COEFFICIENTS, STARTING WITH THE CONSTANT TERM

COEFFICIENT A( 0 ) ? -6
COEFFICIENT A( 1 ) ? 1
COEFFICIENT A( 2 ) ? 1

FOR THE POLYNOMIAL EQUATION:

0 = -6 + 1X + 1X↑2


THE ROOTS ARE:

ROOT #           REAL PART         IMAGINARY PART
 1                2                0
 2               -3                0

READY
>_
```

Fig. 5-2. Results of determining the roots for the second-order polynomial.

```
                REAL & COMPLEX ROOTS OF A REAL POLYNOMIAL

ORDER OF POLYNOMIAL ? 4

ENTER COEFFICIENTS, STARTING WITH THE CONSTANT TERM

COEFFICIENT A( 0 ) ? 1
COEFFICIENT A( 1 ) ? 2.613
COEFFICIENT A( 2 ) ? 3.414
COEFFICIENT A( 3 ) ? 2.613
COEFFICIENT A( 4 ) ? 1

FOR THE POLYNOMIAL EQUATION:

0 =  1  +  2.613 X  +  3.414 X↑ 2  +  2.613 X↑ 3  +  1 X↑ 4


THE ROOTS ARE:

ROOT #           REAL PART         IMAGINARY PART
 1               -.382629          -.923902
 2               -.382629           .923902
 3               -.92387           -.382705
 4               -.92387            .382705

READY
>
```

Fig. 5-3. Results of determining pole locations for the fourth-order Butterworth polynomial.

been factored. Consequently, the poles (i.e., the roots) of the denominator of the transfer function are known, and can be either real or complex. If this is not the case, the PLYROOT program discussed earlier in this chapter solves for both real and complex roots of a polynomial.

When executed, the program requires the number of numerator coefficients and their values, and the number of roots. Each denominator root is assumed to be a complex number and is entered in the format A,B where A and B are the real and imaginary parts, respectively. In addition, all

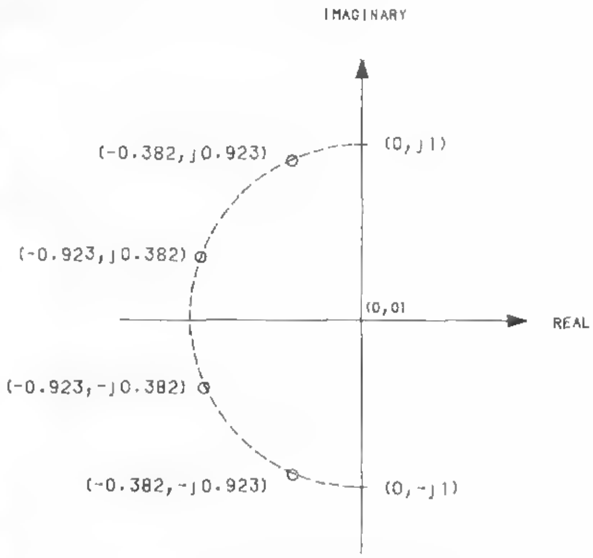


Fig. 5-4. Pole locations for fourth-order Butterworth polynomial.

roots must be distinct, so that multiple roots are not allowed. The computed inverse Laplace transform time response is expressed in the form:

$$f(t) = (a_1 + jb_1)e^{(c_1 + jd_1)t} + (a_2 + jb_2)e^{(c_2 + jd_2)t} + \ldots \quad \text{(Eq. 5-4)}$$

so that there will be as many terms as there are roots.

Example 5-3

Determine the inverse Laplace transform time response for the transfer function

$$T(s) = \frac{s - 1}{(s + 2)\ (s + 1)}$$

There are two coefficients for the numerator and two real denominator roots. Fig. 5-6 shows the

```
100 ' INVERSE LAPLACE TRANSFORM (ILAPLACE)
101 CLS:PRINT,"INVERSE LAPLACE TRANSFORM PROGRAM"
102 DIM A(20),CI(20),CR(20),RI(20),RR(20),SR(20),SI(20),DR(20),DI(20)
103 PRINT:INPUT"NUMBER OF NUMERATOR COEFFICIENTS ";M
104 FOR I=1 TO M
105 PRINT"COEFFICIENT";I;
106 INPUT A(I):NEXTI
107 INPUT"NUMBER OF DENOMINATOR ROOTS ";N
108 FOR I=1 TO N
109 PRINT"ROOT";I;"(REAL, IMAGINARY): ";
110 INPUTRR(I),RI(I)
111 NEXT I
112 FOR I=1 TO N
113 S1=RR(I)
114 S2=RI(I)
115 PR=1
116 PI=0
117 TR=A(1)
118 TI=0
119 IF(M-1)<0 THEN 160  ELSE 120
120 IF  (M-1)>0 THEN 121  ELSE 129
121 FOR J=2 TO M
122 B1=PR*S1-PI*S2
123 C1=PI*S1+PR*S2
124 PR=B1
125 PI=C1
126 TR=TR+A(J)*PR
127 TI=TI+A(J)*PI
128 NEXT J
129 SR(I)=TR
130 SI(I)=TI
131 NEXT I
132 FOR J=1 TO N
133 PR=1
134 PI=0
135 FOR I=1 TO N
136 IF J-I<>0 THEN 137  ELSE 143
```

Continued on next page.

Fig. 5-5. Listing for ILAPLACE program.

```
137 UR=RR(J)-RR(I)
138 UI=RI(J)-RI(I)
139 B1=PR*UR-PI*UI
140 C1=PR*UI+PI*UR
141 PR=B1
142 PI=C1
143 NEXT I
144 DR(J)=PR
145 DI(J)=PI
146 NEXT J
147 FOR I=1 TO N
148 CR(I)=(SR(I)*DR(I)+SI(I)*DI(I))/((DR(I)[2)+(DI(I)[2))
149 CI(I)=(SI(I)*DR(I)-SR(I)*DI(I))/(DR(I)*DR(I)+DI(I)*DI(I))
150 NEXT I
151 I=1
152 PRINT:PRINT"THE INVERSE LAPLACE TRANSFORM, AS A FUNCTION OF TIME IS:":PRINT
153 PRINT"F(T) = <";CR(I);" + J(";CI(I);")>  * EXP<";RR(I);
154 PRINT" + J(";RI(I);")>*T"
155 IF N>1 THEN 156  ELSE 160
156 FOR I=2 TO N
157 PRINT"       +<";CR(I);" + J(";CI(I);")>  * EXP<";RR(I);
158 PRINT" + J(";RI(I);")>*T"
159 NEXT I
160 PRINT:END
```

Fig. 5-5 (cont). Listing for ILAPLACE program.

output results for this transfer function, which has the time variant solution:

$$f(t) = 3e^{-2t} - 2e^{-t}$$

which agrees with the result obtained by the use of inverse Laplace transform tables.

Example 5-4

Determine the inverse Laplace transform time response for the transfer function

$$T(s) = \frac{1}{s^2 + 4s + 5} = \frac{1}{(s + 2 + j)(s + 2 - j)}$$

For this transfer function, there are 2 complex roots, which are complex conjugates. The output results are shown in Fig. 5-7, but can be simplified further.

The computed solution can then be rewritten as:

$$j0.5e^{.5t}e^{-(2+j)t} - j0.5e^{-(2-j)t} =$$
$$e^{-2t}(0.5je^{-t} - 0.5je^{t}) = e^{-2t}\sin(t)$$

using the exponential identities of trigonometric functions.

Example 5-5

Determine the inverse Laplace transform time response for the transfer function:

$$T(s) = \frac{1}{s(s^2 - 1)} = \frac{1}{s(s + 1)(s - 1)}$$

Fig. 5-8 shows the output results, which can be simplified further. Using the identity

$$\frac{e^{t} + e^{-t}}{2} = \cosh(t) \qquad \text{(Eq. 5-6)}$$

so that the computed answer can now be more formally stated as:

$$f(t) = \cosh(t) - 1$$

which can be verified using tables of inverse Laplace transforms.

IMPEDANCE MATCHING PADS

Impedance matching devices are frequently used to "match" the signal source to a given load impedance. These devices may be either transformers, transistor circuits, or a class of networks called pads. Pads may be composed solely of resistors, or they may be a combination of inductors and capacitors. In this section, programs for the design of several common types of resistive and lossless matching pads are discussed.

```
INVERSE LAPLACE TRANSFORM PROGRAM

NUMBER OF NUMERATOR COEFFICIENTS ? 2
COEFFICIENT 1 ? -1
COEFFICIENT 2 ? 1
NUMBER OF DENOMINATOR ROOTS ? 2
ROOT 1 (REAL, IMAGINARY): ? -2,0
ROOT 2 (REAL, IMAGINARY): ? -1,0


THE INVERSE LAPLACE TRANSFORM, AS A FUNCTION OF TIME IS:

F(T) = < 3  + J( 0 )>  * EXP<-2  + J( 0 )>*T
      +<-2  + J( 0 )>  * EXP<-1  + J( 0 )>*T


READY
>_
```

Fig. 5-6. Results for inverse Laplace transfer function (Example 5-3).

```
INVERSE LAPLACE TRANSFORM PROGRAM

NUMBER OF NUMERATOR COEFFICIENTS ? 1
COEFFICIENT 1 ? 1
NUMBER OF DENOMINATOR ROOTS ? 2
ROOT 1 (REAL, IMAGINARY): ? -2,-1
ROOT 2 (REAL, IMAGINARY): ? -2,1


THE INVERSE LAPLACE TRANSFORM, AS A FUNCTION OF TIME IS:

F(T) = < 0  + J( .5 )>  * EXP<-2  + J(-1 )>*T
      +< 0  + J(-.5 )>  * EXP<-2  + J( 1 )>*T

READY
>_
```

Fig. 5-7. Results for inverse Laplace transfer function (Example 5-4).

```
INVERSE LAPLACE TRANSFORM PROGRAM

NUMBER OF NUMERATOR COEFFICIENTS ? 1
COEFFICIENT 1 ? 1
NUMBER OF DENOMINATOR ROOTS ? 3
ROOT 1 (REAL, IMAGINARY): ? 0,0
ROOT 2 (REAL, IMAGINARY): ? -1,0
ROOT 3 (REAL, IMAGINARY): ? 1,0


THE INVERSE LAPLACE TRANSFORM, AS A FUNCTION OF TIME IS:

F(T) = <-1  + J( 0 )>  * EXP< 0  + J( 0 )>*T
      +< .5  + J( 0 )>  * EXP<-1  + J( 0 )>*T
      +< .5  + J( 0 )>  * EXP< 1  + J( 0 )>*T


READY
>_
```

Fig. 5-8. Results for inverse Laplace transfer function (Example 5-5).

Resistance Pads

A resistance pad is a four-terminal resistor network that permits the matching of a signal source to a load, but with a finite amount of loss, or attenuation. This is because resistors are passive devices and, unlike a transistor circuit or amplifier, cannot produce gain. The four main types of resistance pad networks are shown in Fig. 5-9. The T- and Pi-networks are unbalanced pads while the H- and O-pads are their respective balanced counterparts. When the source and load impedances are equal, these networks are referred to as attenuators, otherwise they are called pads when the source and load impedances are not equal.

Fig. 5-10 lists a nongraphics version (ATTNPAD1) of the program to determine the required pad resistance when the source and load impedances, as well as the required amount of attenuation (in decibels) are known. The ATTNPAD2 program in Fig. 5-11 is the same as Fig. 5-10, but graphics are included so that the final schematic is included as part of the video display.

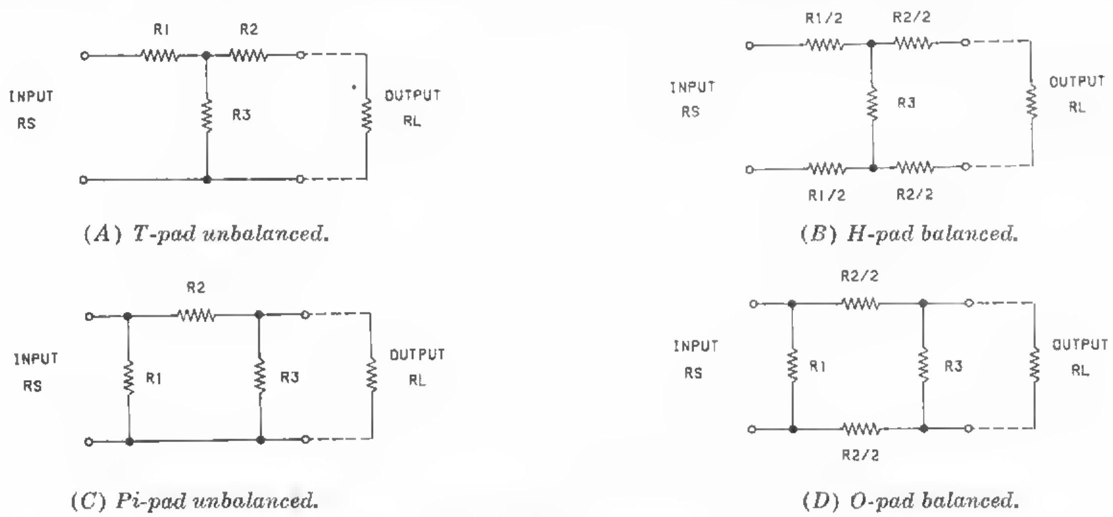
Depending on the relationship between the source and load impedances, there is usually a minimum value of attenuation required in order to satisfy the design. If more attenuation is required, the program informs us of that fact. The ATTNPAD1 program may be easily adapted for use with a line printer, while ATTNPAD2 may not. The following examples use the ATTNPAD2 program.

Example 5-6

Design a T-pad having an attenuation of 10 dB while matching a 300-ohm source to a 75-ohm load. When run, we find that it is not possible to have 10-dB attenuation. Consequently, we must choose a higher value, such as 15 dB. Fig. 5-12 shows the final circuit required.

Example 5-7

Design an O-pad having an attenuation of 5 dB between a 600-ohm source and load. Fig. 5-13 shows the final circuit required.



(*A*) *T-pad unbalanced.* (*B*) *H-pad balanced.*

(*C*) *Pi-pad unbalanced.* (*D*) *O-pad balanced.*

Fig. 5-9. Resistive matching pads.

```
100 'ATTENUATOR PAD DESIGN - WITHOUT GRAPHICS (ATTNPAD1)
101 CLS:PRINT"ATTENUATOR DESIGN:"
102 CLEAR
103 PRINT:PRINT"1. T-PAD (UNBALANCED)"
104 PRINT"2. H-PAD (BALANCED)"
105 PRINT"3. PI-PAD (UNBALANCED)"
106 PRINT"4. O-PAD (BALANCED)"
107 PRINT:INPUT"     SELECT # ";Z9
108 ON Z9 GOTO 109  ,123  ,125  ,139
```

Continued on next page.

Fig. 5-10. Listing for ATTNPAD1 program.

```
109 CLS:PRINT,"UNBALANCED T-PAD DESIGN"
110 GOSUB 147
111 INPUT"ATTENUATION (IN DECIBELS): ";N
112 GOSUB 142
113 ZA=(ZS*C-2*X*D)/Z9:ZA=INT(ZA*10+.5)/10
114 ZB=(ZL*C-2*X*D)/Z9:ZB=INT(ZB*10+.5)/10
115 ZC=2*X*D:ZC=INT(ZC*10+.5)/10
116 IF ZA<0 OR ZB<0 THEN 117  ELSE 120
117 PRINT:PRINT"MORE ATTENUATION IS REQUIRED, TRY AGAIN"
118 FOR F=1 TO 500:NEXT F
119 GOTO 109
120 PRINT:PRINT"THEN:":PRINT"ZA =";ZA;"OHMS":PRINT"ZB =";ZB;"OHMS"
121 PRINT"ZC =";ZC;"OHMS"
122 GOTO 141
123 CLS:PRINT,"BALANCED H-PAD DESIGN"
124 GOTO 110
125 CLS:PRINT,"UNBALANCED PI-PAD DESIGN"
126 GOSUB 147
127 INPUT"ATTENUATION (IN DECIBELS): ";N
128 GOSUB 142
129 ZA=ZS*E:ZA=INT(ZA*10+.5)/10
130 ZB=(X/(D*2))/(Z9-2):ZB=INT(ZB*10+.5)/10
131 ZC=ZL*E:ZC=INT(ZC*10+.5)/10
132 IF ZA<0 OR ZB<0 OR ZC<0 THEN 133  ELSE 136
133 PRINT:PRINT"MORE ATTENUATION IS REQUIRED, TRY AGAIN"
134 FOR F=1 TO 500:NEXT F
135 GOTO 125
136 PRINT:PRINT"THEN:":PRINT"ZA =";ZA;"OHMS":PRINT"ZB =";ZB;"OHMS"
137 PRINT"ZC =";ZC;"OHMS"
138 GOTO 141
139 CLS:PRINT,"BALANCED O-PAD DESIGN"
140 GOTO 126
141 END
142 K2=10↑(N/10):K=SQR(K2)
143 C=(K2+1)/(K2-1):D=K/(K2-1)
144 Z=SQR(ZS/ZL):X=SQR(ZL*ZS)
145 E=(K2-1)/(K2-(2*K/Z)+1)
146 RETURN
147 PRINT:PRINT"INPUT IMPEDANCE: ";
148 INPUT ZS
149 PRINT"LOAD IMPEDANCE : ";
150 INPUT ZL:RETURN
151 END
```

Fig. 5-10. (cont). Listing for ATTNPAD1 program.

```
100 'ATTENUATOR PAD DESIGN - WITH GRAPHICS (ATTNPAD2)
101 CLS:PRINT"ATTENUATOR DESIGN:"
102 CLEAR
103 PRINT:PRINT"1. T-PAD (UNBALANCED)"
104 PRINT"2. H-PAD (BALANCED)"
105 PRINT"3. PI-PAD (UNBALANCED)"
106 PRINT"4. O-PAD (BALANCED)"
107 PRINT:INPUT"      SELECT # ";Z9
108 ON Z9 GOTO 109  ,126  ,128  ,145
109 CLS:PRINT@20,"UNBALANCED T-PAD DESIGN"
110 GOSUB 153
111 INPUT"ATTENUATION (IN DECIBELS): ";N
112 GOSUB 148
113 ZA=(ZS*C-2*X*D)/Z9:ZA=INT(ZA*10+.5)/10
114 ZB=(ZL*C-2*X*D)/Z9:ZB=INT(ZB*10+.5)/10
115 ZC=2*X*D:ZC=INT(ZC*10+.5)/10
```

Continued on next page.

Fig. 5-11. Listing for ATTNPAD2 program.

```
116 IF ZA<0 OR ZB<0 THEN 117  ELSE 120
117 PRINT:PRINT"MORE ATTENUATION IS REQUIRED, TRY AGAIN"
118 FOR F=1 TO 500:NEXT F
119 GOTO 109
120 PRINT@84,ZA:PRINT@100,ZB:PRINT@290,ZC
121 IF Z9=2 THEN 122  ELSE 123
122 PRINT@596,ZA:PRINT@612,ZB
123 GOSUB 158
124 PRINT@832," "
125 GOTO 147
126 CLS:PRINT@21,"BALANCED H-PAD DESIGN"
127 GOTO 110
128 CLS:PRINT@20,"UNBALANCED PI-PAD DESIGN"
129 GOSUB 153
130 INPUT"ATTENUATION (IN DECIBELS): ";N
131 GOSUB 148
132 ZA=ZS*E:ZA=INT(ZA*10+.5)/10
133 ZB=(X/(D*2))/(Z9-2):ZB=INT(ZB*10+.5)/10
134 ZC=ZL*E:ZC=INT(ZC*10+.5)/10
135 IF ZA<0 OR ZB<0 OR ZC<0 THEN 136  ELSE 139
136 PRINT:PRINT"MORE ATTENUATION IS REQUIRED, TRY AGAIN"
137 FOR F=1 TO 500:NEXT F
138 GOTO 128
139 PRINT@92,ZB:PRINT@397,ZA:PRINT@425,ZC
140 IF Z9=4 THEN 141  ELSE 142
141 PRINT@604,ZB
142 GOSUB 179
143 PRINT@832," "
144 GOTO 147
145 CLS:PRINT@21,"BALANCED O-PAD DESIGN"
146 GOTO 129
147 END
148 K2=10↑(N/10):K=SQR(K2)
149 C=(K2+1)/(K2-1):D=K/(K2-1)
150 Z=SQR(ZS/ZL):X=SQR(ZL*ZS)
151 E=(K2-1)/(K2-(2*K/Z)+1)
152 RETURN
153 PRINT@704,"INPUT IMPEDANCE: ";
154 INPUT ZS
155 PRINT@738,"LOAD IMPEDANCE : ";
156 INPUT ZL:RETURN
157 END
158 PRINT@256,"INPUT":PRINT@314,"LOAD"
159 FOR X=30 TO 42:SET(X,7):NEXTX
160 FOR X=50 TO 74:SET(X,7):NEXTX
161 FOR X=82 TO 94:SET(X,7):NEXTX
162 FOR Y=6 TO 8:SET(42,Y):SET(50,Y):SET(74,Y):SET(82,Y):NEXTY
163 FOR X=42 TO 50:SET(X,6):SET(X,8):NEXTX
164 FOR X=74 TO 82:SET(X,6):SET(X,8):NEXTX
165 FOR Y=7 TO 12:SET(62,Y):NEXTY
166 FOR Y=16 TO 22:SET(62,Y):NEXTY
167 FOR X=60 TO 64:SET(X,12):SET(X,16):NEXTX
168 FOR Y=12 TO 16:SET(60,Y):SET(64,Y):NEXTY
169 IF Z9=2 THEN 170  ELSE 177
170 FOR X=30 TO 42:SET(X,22):NEXTX
171 FOR X=50 TO 74:SET(X,22):NEXTX
172 FOR X=82 TO 94:SET(X,22):NEXTX
173 FOR Y=21 TO 23:SET(42,Y):SET(50,Y):SET(74,Y):SET(82,Y):NEXTY
174 FOR X=42 TO 50:SET(X,21):SET(X,23):NEXTX
175 FOR X=74 TO 82:SET(X,21):SET(X,23):NEXTX
176 RETURN
177 FOR X=30 TO 94:SET(X,22):NEXTX
```

Continued on next page.

Fig. 5-11 (cont). Listing for ATTPAD2 program.

```
178 RETURN
179 PRINT@256,"INPUT":PRINT@314,"LOAD"
180 FOR X=30 TO 58:SET(X,7):NEXTX
181 FOR X=66 TO 94:SET(X,7):NEXTX
182 FOR X=58 TO 66:SET(X,6):SET(X,8):NEXTX
183 FOR Y=7 TO 12:SET(44,Y):SET(80,Y):NEXTY
184 FOR Y=16 TO 22:SET(44,Y):SET(80,Y):NEXTY
185 FOR Y=12 TO 16:SET(42,Y):SET(46,Y):SET(78,Y):SET(82,Y):NEXTY
186 FOR X=42 TO 46:SET(X,12):SET(X,16):NEXTX
187 FOR X=78 TO 82:SET(X,12):SET(X,16):NEXTX
188 IF Z9=4 THEN 189  ELSE 194
189 FOR X=30 TO 58:SET(X,22):NEXTX
190 FOR X=66 TO 94:SET(X,22):NEXTX
191 FOR X=58 TO 66:SET(X,21):SET(X,23):NEXTX
192 FOR Y=21 TO 23:SET(58,Y):SET(66,Y):NEXTY
193 RETURN
194 FOR X=30 TO 94:SET(X,22):NEXTX
195 RETURN
```

Fig. 5-11 (cont). Listing for ATTNPAD2 program.

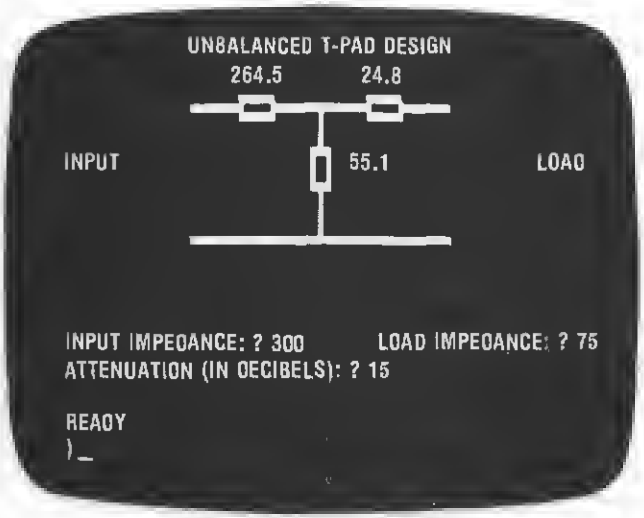


Fig. 5-12. Video results of T-pad having an attenuation of 15dB.

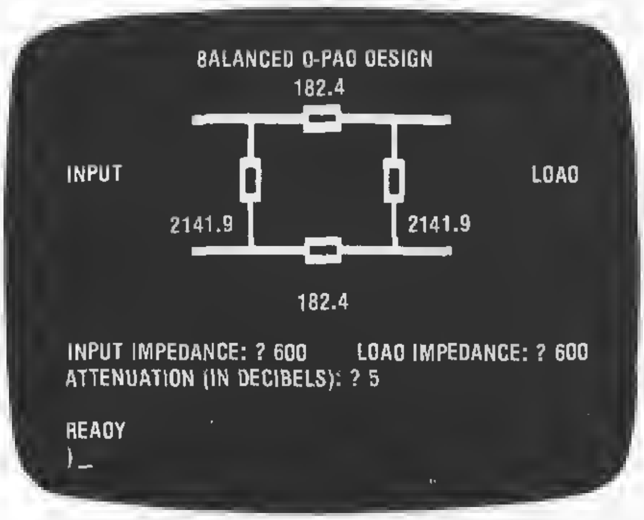
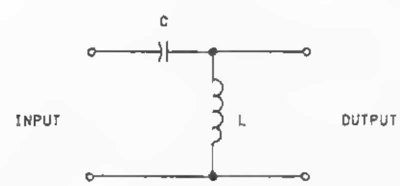


Fig. 5-13. Video results of O-pad having an attenuation of 5dB.

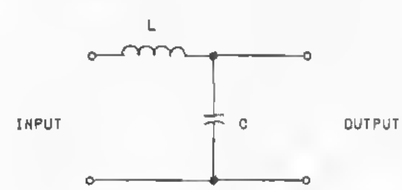
Lossless Pads

Fig. 5-14 shows two types of lossless pads that are used to match impedances at a single frequency. Both circuits are equivalent, and require that the source impedance be less than the load impedance. If the source impedance is greater than the load, then either circuit is turned around so that the shunt element is now in parallel with the source instead of with the load.

Fig. 5-15 lists a nongraphics program (LCPAD1) which determines the required series capacitor/shunt inductor circuit (Fig. 5-14A), or the equivalent series inductor/shunt capacitor circuit (Fig. 5-14B) when the source and load im-



(*A*) *Series capacitor/shunt inductor.*



(*B*) *Series inductor/shunt capacitor.*

Fig. 5-14. LC lossless pads for a specific frequency.

pedances, as well as the input frequency, are known. The LCPAD2 program of Fig. 5-16 includes a schematic which is included as part of the video display.

Example 5-8

Design a lossless pad to match a 50-ohm source to a 75-ohm load at 3.5 MHz.

```
100 'LOSSLESS L-C PAD DESIGN - WITHOUT GRAPHICS (LCPAD1)
101 CLS:PRINT,"LOSSLESS L-C PAD DESIGN"
102 PRINT:INPUT"INPUT IMPEDANCE ";Z1
103 INPUT"LOAD IMPEDANCE ";Z2
104 INPUT"INPUT FREQUENCY (IN HZ) ";F
105 W=F/6.28318
106 X1=Z2*SQR(Z1/(Z2-Z1)):X2=SQR(Z1*(Z2-Z1))
107 S1=1*1E9/(W*X2):H1=X1*1E6/W
108 S1=INT(S1*100+.5)/100:H1=INT(H1*100+.5)/100
109 S2=X2*1E6/W:H2=1*1E9/(W*X1)
110 S2=INT(S2*100+.5)/100:H2=INT(H2*100+.5)/100
111 PRINT:PRINT"THEN, EITHER:"
112 PRINT:PRINT"SHUNT INDUCTANCE =";H1;"UH"
113 PRINT"SERIES CAPACITANCE =";S1;"NF"
114 PRINTTAB(6)"OR:"
115 PRINT:PRINT"SHUNT CAPACITANCE =";H2;"NF"
116 PRINT"SERIES INDUCTANCE =";S2;"UH"
117 END
```

Fig. 5-15 Listing for LCPAD1 program.

```
100 'LOSSLESS L-C PAD DESIGN - GRAPHICS (LCPAD2)
101 CLS:PRINT@20,"LOSSLESS L-C PAD DESIGN"
102 PRINT@640,"INPUT IMPEDANCE ";
103 INPUT Z1
104 PRINT@673,"LOAD IMPEDANCE ";
105 INPUT Z2
106 INPUT"INPUT FREQUENCY (IN HZ) ";F
107 W=F/6.28318
108 X1=Z2*SQR(Z1/(Z2-Z1)):X2=SQR(Z1*(Z2-Z1))
109 S1=1*1E9/(W*X2):H1=X1*1E6/W
110 S1=INT(S1*100+.5)/100:H1=INT(H1*100+.5)/100
111 S2=X2*1E6/W:H2=1*1E9/(W*X1)
112 S2=INT(S2*100+.5)/100:H2=INT(H2*100+.5)/100
113 PRINT@73,S1;"NF":PRINT@115,S2;"UH"
114 PRINT@403,H1;"UH":PRINT@431,H2;"NF"
115 PRINT@286,"OR"
116 GOSUB 119
117 PRINT@832," "
118 END
119 FOR X=0 TO 16:SET(X,7):NEXT
120 FOR X=18 TO 50:SET(X,7):NEXT
121 FOR Y=6 TO 8:SET(16,Y):SET(18,Y):NEXT
122 FOR X=0 TO 50:SET(X,22):NEXT
123 FOR Y=7 TO 11:SET(36,Y):NEXT
124 SET(37,11):SET(38,11):SET(37,17):SET(38,17)
125 FOR Y=11 TO 17:SET(38,Y):NEXT
126 FOR X=34 TO 37:SET(X,13):SET(X,15):NEXT
127 FOR Y=17 TO 22:SET(36,Y):NEXT
128 FOR X=76 TO 89:SET(X,7):NEXT
129 SET(89,6):SET(103,6):FOR X=90 TO 102:SET(X,6):NEXT
130 FOR Y=6 TO 8:SET(93,Y):SET(94,Y):SET(98,Y):SET(99,Y):NEXT
131 FOR X=103 TO 127:SET(X,7):NEXT
132 FOR Y=7 TO 13:SET(113,Y):NEXT
133 FOR Y=15 TO 22:SET(113,Y):NEXT
134 FOR X=111 TO 115:SET(X,13):SET(X,15):NEXT
135 FOR X=76 TO 127:SET(X,22):NEXT
136 RETURN
```

Fig. 5-16. Listing for LCPAD2 program.

The two possible circuits are shown in Fig. 5-17, along with the required values.

Another version of the lossless pad is to be able to specify the phase shift in electrical degrees. Depending on the source and load impedances, and the amount of phase shift required, there are four circuits possible, as shown in Fig. 5-18. The convention is taken that if the input signal is to lead the output, the phase angle is positive; otherwise, it is negative if the input lags the output.

The LCPSPAD1 program of Fig. 5-19 is a nongraphic version for designing lossless pads, while Fig. 5-20 is the graphics version which includes the schematic with component values as part of the video display.

Example 5-9

Design a lossless pad to match a 50-ohm source with a 75-ohm load with a phase shift of +45 degrees at 3.5 MHz. Repeat this example, but for a phase shift of −45 degrees.

The final circuit for +45 degrees phase shift is shown in Fig. 5-21, while the display of Fig. 5-22 shows the required circuit having a phase shift of −45 degrees.

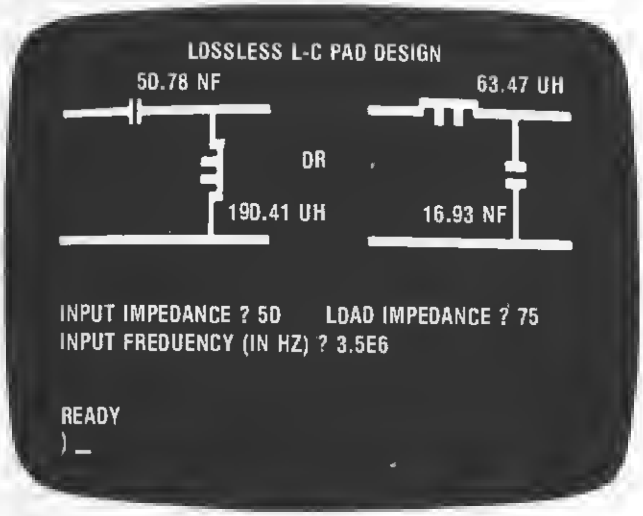


Fig. 5-17. Video results of lossless LC pad design.

PI-TEE (DELTA-WYE) TRANSFORM

Networks often topologically appear in forms resembling either "Tee" or "Pi" shaped networks (sometimes called Wye or Delta networks), as shown in Fig. 5-23. Examples are the resistive attenuators discussed previously, Wheatstone bridge, and lattice circuits. The ability to transform from Tee to Pi, or vice versa, simplifies the analysis of a number of circuits.

The TEEPI program listed in Fig. 5-24 allows the user to determine the corresponding equivalent network. The starting and equivalent schematics are displayed on the screen along with their values.

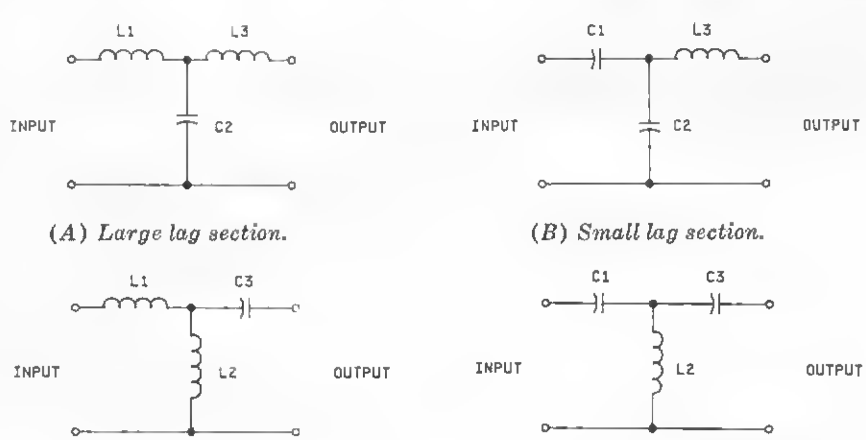
Example 5-10

For the unbalanced T-pad design of Example 5-6, determine the equivalent Pi (or Delta) circuit.

From the results of Example 5-6, Z1, Z2, and Z3 are 264.5, 55.1, and 24.8 ohms, respectively. The equivalent Pi circuit is shown in Fig. 5-25. In fact, the resulting Pi circuit should be the same as if we had chosen to design an unbalanced Pi-pad in Example 5-6 instead of a T-pad.

MESH CURRENT NETWORK ANALYSIS

In an electric network, mesh currents (or loop currents) are produced in each circuit branch as



(A) Large lag section. *(B) Small lag section.*

(C) Large lead section. *(D) Small lead section.*

Fig. 5-18. LC lossless pads for use for a specific phase shift and frequency.

```
100 'LOSSLESS L-C PAD WITH PHASE SHIFT - W/O GRAPHICS (LCPSPAD1)
101 CLS:PRINT,"LOSSLESS L-C PAD DESIGN"
102 PRINT:INPUT"INPUT IMPEDANCE ";Z1
103 INPUT"LOAD IMPEDANCE ";Z2
104 INPUT"INPUT FREQUENCY (IN HZ) ";F
105 PRINT"REQUIRED PHASE SHIFT (IN DEGREES) "
106 INPUT"  (+/- ANGLE, INPUT SIGNAL LEAD/LAGS OUTPUT) ";D
107 PRINT:PRINT"THEN:"
108 W=F/6.28319:R=D/57.2957
109 Z=Z1/Z2:Y=SQR(Z)
110 A=Z*SIN(R)/(Y-COS(R))
111 B=Y*SIN(R)
112 C=Y*SIN(R)/(1-Y*COS(R))
113 IF R<0 THEN 115  ELSE 114
114 IF R>0 THEN 135  ELSE 113
115 X3=Z1/ABS(B):L3=X3*1E6/W
116 X2=Z1/ABS(A):C2=1E9/(W*X2)
117 L3=INT(L3*100+.5)/100
118 C2=INT(C2*100+.5)/100
119 IF C<0 THEN 121  ELSE 120
120 IF C>0 THEN 129  ELSE 119
121 X1=Z1/ABS(C):C1=1E9/(W*X1)
122 C1=INT(C1*100+.5)/100
123 C2=INT(C2*100+.5)/100
124 L3=INT(L3*100+.5)/100
125 PRINT"ELEMENT 1 (CAPACITOR) =";C1;"NF"
126 PRINT"ELEMENT 2 (CAPACITOR) =";C2;"NF"
127 PRINT"ELEMENT 3 (INDUCTOR) =";L3;"UH"
128 GOTO 151
129 X1=Z1/C:L1=X1*1E6/W
130 L1=INT(L1*100+.5)/100
131 PRINT"ELEMENT 1 (INDUCTOR) =";L1;"UH"
132 PRINT"ELEMENT 2 (CAPACITOR) =";C2;"NF"
133 PRINT"ELEMENT 3 (INDUCTOR) =";L3;"UH"
134 GOTO 151
135 X2=Z1/A:L2=X2*1E6/W:X3=Z1/B:C3=1E9/(W*X3)
136 L2=INT(L2*100+.5)/100
137 C3=INT(C3*100+.5)/100
138 IF C<0 THEN 140  ELSE 139
139 IF C>0 THEN 146  ELSE 138
140 X1=Z1/ABS(C):C1=1E9/(X1*W)
141 C1=INT(C1*100+.5)/100
142 PRINT"ELEMENT 1 (CAPACITOR) =";C1;"NF"
143 PRINT"ELEMENT 2 (INDUCTOR) =";L2;"UH"
144 PRINT"ELEMENT 3 (CAPACITOR) =";C3;"NF"
145 GOTO 151
146 X1=Z1/C:L1=X1*1E6/W
147 L1=INT(L1*100+.5)/100
148 PRINT"ELEMENT 1 (INDUCTOR) =";L1;"UH"
149 PRINT"ELEMENT 2 (INDUCTOR) =";L2;"UH"
150 PRINT"ELEMENT 3 (CAPACITOR) =";C3;"NF"
151 PRINT:PRINT:END
```

Fig. 5-19. Listing for LCPSPAD1 program.

```
129 L1=INT(L1*100+.5)/100
130 PRINT@205,L1;"UH":PRINT@234,L3;"UH"
131 PRINT@256,"INPUT":PRINT@314,"OUTPUT":PRINT@354,C2;"NF"
132 GOSUB 159  :GOTO 148
133 X2=Z1/A:L2=X2*1E6/W:X3=Z1/B:C3=1E9/(W*X3)
134 L2=INT(L2*100+.5)/100
135 C3=INT(C3*100+.5)/100
136 IF C<0 THEN 138  ELSE 137
137 IF C>0 THEN 143  ELSE 136
138 X1=Z1/ABS(C):C1=1E9/(X1*W)
139 C1=INT(C1*100+.5)/100
140 PRINT@202,C1;"NF":PRINT@238,C3;"NF"
141 PRINT@256,"INPUT":PRINT@314,"OUTPUT":PRINT@354,L2;"UH"
142 GOSUB 170  :GOTO 148
143 X1=Z1/C:L1=X1*1E6/W
144 L1=INT(L1*100+.5)/100
145 PRINT@205,L1;"UH":PRINT@238,C3;"NF"
146 PRINT@256,"INPUT":PRINT@314,"OUTPUT":PRINT@356,L2;"UH"
147 GOSUB 180
148 PRINT@832," ":END
149 FOR X=10 TO 36:SET(X,7):NEXT
150 FOR X=39 TO 83:SET(X,7):NEXT
151 FOR X=98 TO 117:SET(X,7):NEXT
152 FOR Y=6 TO 8:SET(36,Y):SET(39,Y):NEXT
153 FOR X=83 TO 98:SET(X,6):NEXT
154 FOR Y=6 TO 8:SET(88,Y):SET(87,Y):SET(93,Y):SET(94,Y):NEXT
155 FOR Y=7 TO 13:SET(64,Y):NEXT
156 FOR Y=15 TO 22:SET(64,Y):NEXT
157 FOR X=62 TO 66:SET(X,13):SET(X,15):NEXT
158 FOR X=10 TO 117:SET(X,22):NEXT:RETURN
159 FOR X=10 TO 30:SET(X,7):NEXT
160 FOR X=45 TO 83:SET(X,7):NEXT
161 FOR X=98 TO 117:SET(X,7):NEXT
162 FOR X=30 TO 45:SET(X,6):NEXT
163 FOR X=83 TO 98:SET(X,6):NEXT
164 FOR Y=6 TO 8:SET(35,Y):SET(34,Y):SET(40,Y):SET(41,Y):NEXT
165 FOR Y=6 TO 8:SET(87,Y):SET(88,Y):SET(93,Y):SET(94,Y):NEXT
166 FOR X=10 TO 117:SET(X,22):NEXT
167 FOR Y=7 TO 13:SET(64,Y):NEXT
168 FOR Y=15 TO 22:SET(64,Y):NEXT
169 FOR X=62 TO 66:SET(X,13):SET(X,15):NEXT:RETURN
170 FOR X=10 TO 36:SET(X,7):NEXT
171 FOR X=39 TO 90:SET(X,7):NEXT
172 FOR X=93 TO 117:SET(X,7):NEXT
173 FOR Y=6 TO 8:SET(36,Y):SET(39,Y):SET(90,Y):SET(93,Y):NEXT
174 FOR X=10 TO 117:SET(X,22):NEXT
175 FOR Y=7 TO 11:SET(64,Y):NEXT
176 FOR Y=17 TO 22:SET(64,Y):NEXT
177 SET(65,11):SET(65,17)
178 FOR Y=11 TO 17:SET(66,Y):NEXT
179 FOR X=62 TO 66:SET(X,13):SET(X,15):NEXT:RETURN
180 FOR X=10 TO 30:SET(X,7):NEXT
181 FOR X=45 TO 90:SET(X,7):NEXT
182 FOR X=93 TO 117:SET(X,7):NEXT
183 FOR X=30 TO 45:SET(X,6):NEXT
184 FOR Y=6 TO 8:SET(35,Y):SET(36,Y):SET(40,Y):SET(41,Y):NEXT
185 FOR Y=6 TO 8:SET(90,Y):SET(93,Y):NEXT
186 FOR X=10 TO 117:SET(X,22):NEXT
187 FOR Y=7 TO 11:SET(64,Y):NEXT
188 FOR Y=17 TO 22:SET(64,Y):NEXT
189 FOR Y=11 TO 17:SET(66,Y):SET(67,Y):NEXT
190 FOR X=62 TO 65:SET(X,15):SET(X,13):NEXT
191 SET(65,11):SET(65,17):RETURN
```

Continued on next page.

Fig. 5-20. Listing for LCPSPAD2 program.

```
100 'LOSSLESS L-C PAD WITH PHASE SHIFT - W/GRAPHICS (LCPSPAD2)
101 CLS:PRINT@20,"LOSSLESS L-C PAD DESIGN"
102 PRINT@576,"INPUT IMPEDANCE ";
103 INPUT Z1
104 PRINT@611,"LOAD IMPEDANCE ";
105 INPUT Z2
106 INPUT"INPUT FREQUENCY (IN HZ) ";F
107 PRINT"REQUIRED PHASE SHIFT (IN DEGREES) "
108 INPUT"  (+/- ANGLE, INPUT SIGNAL LEADS/LAGS OUTPUT) ";D
109 W=F/6.28319:R=D/57.2957
110 Z=Z1/Z2:Y=SQR(Z)
111 A=Z*SIN(R)/(Y-COS(R))
112 B=Y*SIN(R)
113 C=Y*SIN(R)/(1-Y*COS(R))
114 IF R<0 THEN 116  ELSE 115
115 IF R>0 THEN 133  ELSE 114
116 X3=Z1/ABS(B):L3=X3*1E6/W
117 X2=Z1/ABS(A):C2=1E9/(W*X2)
118 L3=INT(L3*100+.5)/100
119 C2=INT(C2*100+.5)/100
120 IF C<0 THEN 122  ELSE 121
121 IF C>0 THEN 128  ELSE 120
122 X1=Z1/ABS(C):C1=1E9/(W*X1)
123 C1=INT(C1*100+.5)/100
124 L3=INT(L3*100+.5)/100
125 PRINT@205,C1;"NF":PRINT@236,L3;"UH"
126 PRINT@256,"INPUT":PRINT@314,"OUTPUT":PRINT@354,C2;"NF"
127 GOSUB 149   :GOTO 148
128 X1=Z1/C:L1=X1*1E6/W
```

Fig. 5-20 (cont). Listing for LCPSPAD2 program.

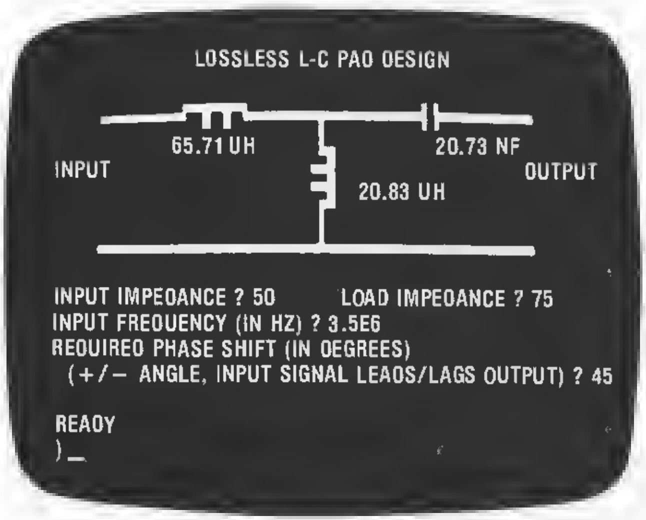


Fig. 5-21. Video results of LC lossless pad for 45 degrees phase shift.

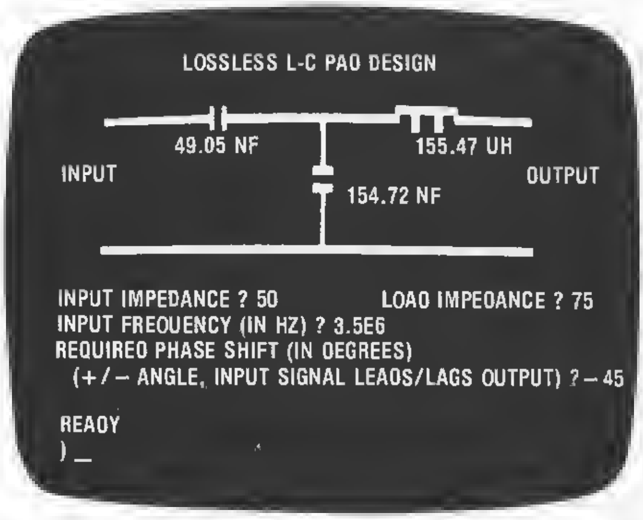


Fig. 5-22. Video results of LC lossless pad for –45 degrees phase shift.

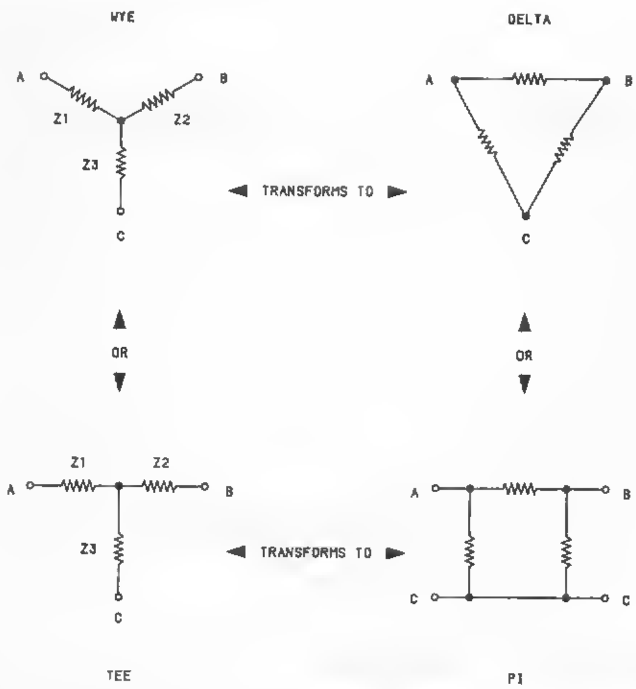


Fig. 5-23. Pi-Tee and Delta-Wye equivalent networks.

```
100 'PI-TEE (DELTA-WYE) TRANSFORM - TEEPI
101 CLS:PRINT,"PI-TEE (DELTA-WYE) TRANSFORMATION"
102 PRINT:PRINT:PRINT"SELECT:"
103 PRINT"      1. <T>EE ";CHR$(94);" PI"
104 PRINT"      2. <P>I ";CHR$(94);" TEE"
105 INPUT"            CHOICE ";C$
106 IF C$<>"T" AND C$<>"P" THEN 105
107 IF C$="T" THEN 108  ELSE 120
108 CLS:PRINT,"TEE-TO-PI TRANSFORMATION"
109 GOSUB 154  :PRINT@286," TO >":GOSUB 155  :GOSUB 133
110 PRINT@576,"Z1 ";
111 INPUTZ1:PRINT@603,"Z2 ";
112 INPUTZ2:PRINT@624,"Z3 ";
113 INPUT Z3
114 ZA=Z3+Z1+(Z3*Z1/Z2)
115 ZB=Z1+Z2+(Z1*Z2/Z3)
116 ZC=Z2+Z3+(Z2*Z3/Z1)
117 PRINT"THEN:":PRINT
118 PRINT@768,"ZA =";ZA:PRINT@795,"ZB =";ZB:PRINT@816,"ZC =";ZC
119 PRINT:END
120 CLS:PRINT,"PI-TO-TEE TRANSFORMATION"
121 GOSUB 154  :PRINT@286,"< TO":GOSUB 155  :GOSUB 133
122 PRINT@576,"ZA ";
123 INPUTZA:PRINT@603,"ZB ";
124 INPUTZB:PRINT@624,"ZC ";
125 INPUT ZC
126 Z=ZA+ZB+ZC
127 Z1=ZB*ZA/Z
128 Z2=ZB*ZC/Z
129 Z3=ZA*ZC/Z
130 PRINT"THEN:":PRINT
131 PRINT@768,"Z1 =";Z1:PRINT@795,"Z3 =";Z3:PRINT@816,"Z2 =";Z2
132 PRINT:END
133 FOR X=0 TO 8:SET(X,7):NEXT
```

Continued on next page.

Fig. 5-24. Listing for TEEPI program.

```
134 FOR X=16 TO 35:SET(X,7):NEXT
135 FOR X=43 TO 50:SET(X,7):NEXT
136 FOR X=0 TO 50:SET(X,22):NEXT
137 FOR X=8 TO 16:SET(X,6):SET(X,8):NEXT
138 FOR X=35 TO 43:SET(X,6):SET(X,8):NEXT
139 FOR Y=7 TO 12:SET(25,Y):NEXT
140 FOR Y=16 TO 22:SET(25,Y):NEXT
141 FOR X=23 TO 27:SET(X,12):SET(X,16):NEXT
142 FOR Y=12 TO 16:SET(23,Y):SET(27,Y):NEXT
143 FOR X=76 TO 97:SET(X,7):NEXT
144 FOR X=105 TO 127:SET(X,7):NEXT
145 FOR X=76 TO 127:SET(X,22):NEXT
146 FOR X=97 TO 105:SET(X,6):SET(X,8):NEXT
147 FOR Y=6 TO 8:SET(97,Y):SET(105,Y):NEXT
148 FOR Y=7 TO 12:SET(90,Y):SET(113,Y):NEXT
149 FOR Y=16 TO 22:SET(90,Y):SET(113,Y):NEXT
150 FOR X=88 TO 92:SET(X,12):SET(X,16):NEXT
151 FOR X=111 TO 115:SET(X,12):SET(X,16):NEXT
152 FOR Y=12 TO 16:SET(88,Y):SET(92,Y):SET(111,Y):SET(115,Y):NEXT
153 RETURN
154 PRINT@197,"Z1":PRINT@211,"Z3":PRINT@242,"ZA":RETURN
155 PRINT@335,"Z2":PRINT@361,"ZC":PRINT@379,"ZB":RETURN
```

Fig. 5-24 (cont). Listing for TEEPI program.

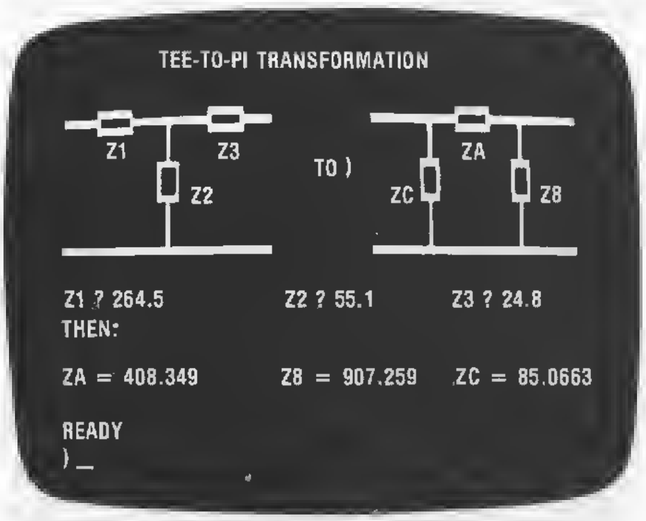


Fig. 5-25. Video results of Tee-to-Pi transformation.

a result of one or more applied voltage sources. From these mesh currents, one can then determine the voltage across or the current through any network element.

Once the mesh equations are established, the individual mesh currents are found by solving a set of simultaneous equations. Fig. 5-26 shows the SIMULEQU program listing to solve a set of simultaneous equations having real coefficients, if a solution is possible. When run, the program requires the number of equations, which is the same number of unknown variables and the minimum allowable value for any solution, which usually is set at 1E–6 (10^{-6}). The coefficients are then entered, one at a time for each equation, with the constant term entered last. For the equation:

$$9X - 8Y + 3Z = 5$$

the "9" is the first coefficient, while the "5" is the fourth coefficient.

```
100 'SOLUTION OF SIMULTANEOUS EQUATIONS WITH REAL COEFFICIENTS (SIMULEQU)
101 CLEAR:CLS
102 DIM A(10,11),X(10)
103 PRINT"SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS"
104 PRINT:INPUT"NUMBER OF EQUATIONS/VARIABLES ";N
105 INPUT"MINIMUM VALUE FOR ANY COEFFICIENT ";MV
106 M=N+1
107 FOR I=1TON
108 PRINT"COEFFICENTS FOR EQUATION";I
109 FORJ=1TOM
110 INPUT"      ";A(I,J)
111 NEXTJ
```

Continued on next page.

Fig. 5-26. Listing for SIMULEQU program.

```
112 NEXT I
113 KK=0:JJ=0
114 FOR I=1TON
115 JJ=KK+1:LL=JJ:KK=KK+1
116 IF (ABS(A(JJ,KK))-MV)<=0 THEN 117  ELSE 119
117 JJ=JJ+1
118 GOTO116
119 IF (LL-JJ)=0 THEN 125  ELSE 120
120 FOR MM=1TOM
121 Z=A(LL,MM)
122 A(LL,MM)=A(JJ,MM)
123 A(JJ,MM)=Z
124 NEXT MM
125 D=A(I,I)
126 FOR J=I TO M
127 A(I,J)=A(I,J)/D
128 NEXTJ
129 K=I+1
130 IF(K-M)<0 THEN 131  ELSE 138
131 FORL=KTON
132 Y=A(L,I)
133 FOR J=I TO M
134 A(L,J)=A(L,J)-A(I,J)*Y
135 NEXTJ
136 NEXTL
137 NEXTI
138 X(N)=A(N,M)
139 L=N
140 FOR J=2TON
141 S=0
142 I=M+1-J
143 FOR K=ITON
144 S=S+A(I-1,K)*X(K)
145 NEXTK
146 L=L-1
147 X(L)=A(I-1,M)-S
148 NEXTJ
149 PRINT:PRINT"THE SOLUTIONS ARE:"
150 FORI=1TON
151 PRINT"X(";I;") =",X(I)
152 NEXTI
153 END
```

Fig. 5-26 (cont). Listing for SIMULEQU program.

Example 5-11

Determine the mesh currents I_a, I_b, and I_c for the network shown in Fig. 5-27, whose three mesh equations can be shown to be written as:

$$(30)I_a - (10)I_b - (20)I_c = 10$$
$$-(10)I_a + (90)I_b - (50)I_c = 0$$
$$-(20)I_a - (50)I_b + (110)I_c = 0$$

Fig. 5-28 shows the computed solutions for these equations so that I_a, I_b, and I_c are 0.477419, 0.135484, and 0.148387 amps, respectively. It is then an easy manner to apply Ohm's law to determine the current through or the voltage across each resistor. For example, the current through the 20-ohm resistor is 0.477419 − 0.148387, or 0.329032 amps, while the voltage drop across it is (0.329032)(20), or 6.58 volts.

Example 5-12

Using the network of Fig. 5-29, determine the five branch currents using Kirchhoff's current and voltage laws.

At nodes A and B, application of the current law gives the two equations:

$$I_1 - I_2 - I_3 = 0$$
$$I_3 - I_4 - I_5 = 0$$

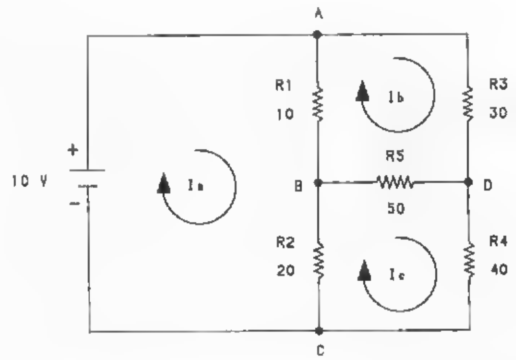
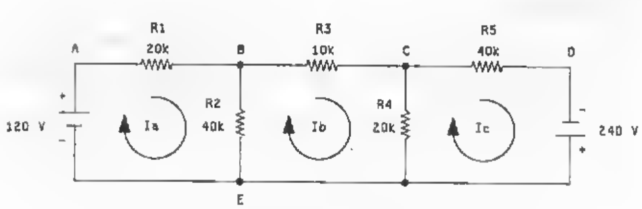


Fig. 5-27. Mesh currents network.

```
SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

NUMBER OF EQUATIONS/VARIABLES ? 3
MINIMUM VALUE FOR ANY SOLUTION ? 1E-6
COEFFICENTS FOR EQUATION 1 :
# 1 ? 30
# 2 ? -10
# 3 ? -20
# 4 ? 10
COEFFICENTS FOR EQUATION 2 :
# 1 ? -10
# 2 ? 90
# 3 ? -50
# 4 ? 0
COEFFICENTS FOR EQUATION 3 :
# 1 ? -20
# 2 ? -50
# 3 ? 110
# 4 ? 0


THE SOLUTIONS ARE:
X( 1 ) = .477419
X( 2 ) = .135484
X( 3 ) = .148387

READY
>_
```

Fig. 5-28. Computed solutions for mesh currents.



LOOP CURRENTS Ia, Ib, AND Ic ARE ARBITRARILY TAKEN AS CLOCKWISE

Fig. 5-29. Circuit for Example 5-12.

while the voltage law yields the following three equations:

$$-(20000)I_4 + (40000)I_5 = 240$$
$$-(40000)I_2 + (10000)I_3 + (20000)I_4 = 0$$
$$(20000)I_1 + (40000)I_2 = 120$$

so that there are five equations with five unknown currents. Fig. 5-30 shows the five computed solutions for this network.

```
SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

NUMBER OF EQUATIONS/VARIABLES ? 5
MINIMUM VALUE FOR ANY SOLUTION ? 1E-6
COEFFICENTS FOR EQUATION 1 :
# 1 ? 1
# 2 ? -1
# 3 ? -1
# 4 ? 0
# 5 ? 0
# 6 ? 0
COEFFICENTS FOR EQUATION 2 :
# 1 ? 0
# 2 ? 0
# 3 ? 1
# 4 ? -1
# 5 ? -1
# 6 ? 0
COEFFICENTS FOR EQUATION 3 :
# 1 ? 0
# 2 ? 0
# 3 ? 0
# 4 ? -20000
# 5 ? 40000
# 6 ? 240
COEFFICENTS FOR EQUATION 4 :
# 1 ? 0
# 2 ? -40000
# 3 ? 10000
# 4 ? 20000
# 5 ? 0
# 6 ? 0
COEFFICENTS FOR EQUATION 5 :
# 1 ? 20000
# 2 ? 40000
# 3 ? 0
# 4 ? 0
# 5 ? 0
# 6 ? 120


THE SOLUTIONS ARE:
X( 1 ) = 4.90909E-03
X( 2 ) = 5.45454E-04
X( 3 ) = 4.36364E-03
X( 4 ) =-1.09091E-03
X( 5 ) = 5.45455E-03


READY
>_
```

Fig. 5-30. Computed solutions for branch currents.

The SIMULEQU program is not valid for complex coefficients, which results from circuits having capacitors and/or inductors in addition to resistors and voltage sources. However, simultaneous equations with complex coefficients are solved using the CPLXSEQ program listed in Fig. 5-31.

When run, the program requires the number of equations, which is the same number of unknown variables, and the minimum allowable value for any solution, which usually is set at 1E − 6 (10^{-6}). The real and imaginary parts of each coefficient are then entered, one at a time for each equation, with the constant term entered last, as shown in the following example.

Example 5-13

For the circuit shown in Fig. 5-32, determine the three loop currents. Writing Kirchhoff's voltage law in rectangular form, we have:

$$(7+3j)I_1 - (5j)I_2 - (5)I_3 = 10/0 = 10$$
$$-(5j)I_1 + (12+3j)I_2 - (2-2j)I_3 = -5/30 = 4.33 - 2.5j$$
$$-(5)I_1 - (2-2j)I_2 + (17-2j)I_3 = -10/90 = -10j$$

Fig. 5-33 shows the computed results that were obtained.

```
100 'SIMULATANEOUS EQUATIONS WITH COMPLEX COEFFICENTS (CPLXSEQ)
101 CLS: DIM A(10,22),B(20,22),P(10)
102 INPUT"NUMBER OF EQUATIONS ";N
103 INPUT"MINIMUM VALUE FOR ANY SOLUTION, ";EP
104 MA=2*N-1:MB=2*N+1:MC=2*N:MD=2*N+2
105 PRINT:PRINT"ENTER:"
106 FOR I=1 TO N
107 PRINT"FOR EQUATION #";I
108 FOR J=1 TO MD STEP 2
109 IF J=MD-1 THEN 112
110 PRINT"    COEFFICIENT";INT(J/2)+1;": REAL, IMAGINARY ";
111 GOTO 113
112 PRINT"    CONSTANT TERM";INT(J/2)+1;": REAL, IMAGINARY ";
113 INPUT A(I,J),A(I,J+1)
114 NEXT J
115 NEXT I
116 IK=-1
117 FOR I=1 TO MA STEP 2
118 IK=IK+1
119 FOR J=1 TO MB STEP 2
120 LJ=I-IK
121 B(I,J)=A(LJ,J):B(I,J+1)=-A(LJ,J+1):NEXT J
122 NEXT I
123 FOR I=2 TO MC STEP 2
124 FOR J=1 TO MB STEP 2
125 LJ=I/2:B(I,J)=A(LJ,J+1):B(I,J+1)=A(LJ,J):NEXT J
126 NEXT I
127 JJ=1
128 FOR I=1 TO MC
129 KK=I-1
130 IF ABS(B(I,I))-EP<>0 THEN 137
131 KK=KK+1:LL=0
132 LL=LL+1
133 FOR J=1 TO MB
134 BT=B(KK,J):KL=KK+LL:B(KK,J)=B(KL,J)
135 B(KL,J)=BT:NEXT J
136 IF ABS(B(I,I))-EP=0 THEN 132  ELSE 137
137 BT=B(I,I)
138 FOR K=1 TO MB:B(I,K)=B(I,K)/BT:NEXT K
139 JJ=JJ+1
140 FOR J=JJ TO MC:BT=B(J,I)
141 FOR K=1 TO MB:B(J,K)=B(J,K)-BT*B(I,K):NEXT K
142 NEXT J
143 NEXT I
```

Continued on next page.

Fig. 5-31. Listing for CPLXSEQ program.

```
144 B(MC,MB)=B(MC,MB)/B(MC,MC):B(MC,MC)=1:JJ=1:KK=2
145 FOR I=1 TO JJ
146 BT=B(I,KK)
147 FOR J=KK TO MB
148 B(I,J)=B(I,J)-BT*B(KK,J):NEXT J
149 NEXT I
150 JJ=JJ+1:KK=KK+1
151 IF JJ-MC<>0 THEN 145  ELSE 152
152 K=0
153 CLS:PRINT"THE SOLUTIONS ARE:"
154 PRINT:PRINTTAB(11)"REAL";TAB(26)"IMAGINARY";
155 PRINTTAB(41)"MAGNITUDE";TAB(56)"PHASE"
156 FOR I=1 TO MC STEP 2
157 K=K+1
158 X(K)=SQR((B(I+1,MB)↑2)+(B(I,MB)↑2))
159 PA=ATN(B(I+1,MB)/B(I,MB))*57.92577
160 IF B(I,MB)<0 THEN 161  ELSE 162
161 P(K)=PA+180:GOTO 163
162 P(K)=PA
163 P(K)=INT(P(K)*100+.5)/100
164 PRINT"X(";K;")";TAB(10)B(I,MB);TAB(25)B(I+1,MB);TAB(40)X(K);TAB(55)P(K)
165 NEXT I
166 PRINT:END
```

Fig. 5-31. (cont). Listing for CPLXSEQ program.

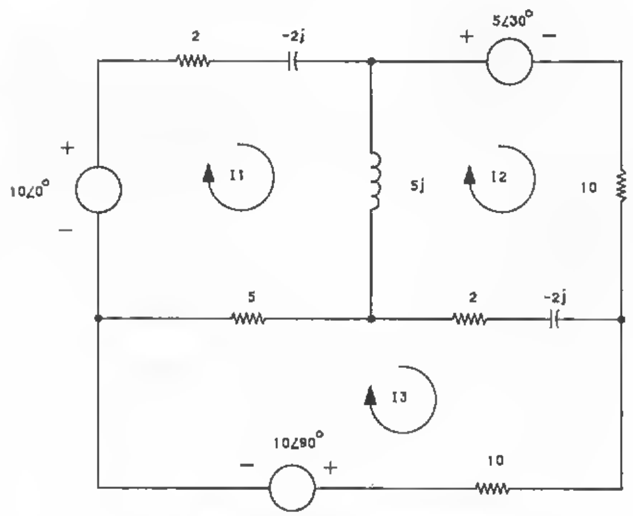


Fig. 5-32. Loop currents network.

```
NUMBER OF EQUATIONS ? 3
MINIMUM VALUE FOR ANY SOLUTION ? 1E-6

ENTER:
FOR EQUATION # 1
    COEFFICIENT 1 : REAL, IMAGINARY ? 7,3
    COEFFICIENT 2 : REAL, IMAGINARY ? 0,-5
    COEFFICIENT 3 : REAL, IMAGINARY ? 5,0
    CONSTANT TERM 4 : REAL, IMAGINARY ? 10,0
FOR EQUATION # 2
    COEFFICIENT 1 : REAL, IMAGINARY ? 0,-5
    COEFFICIENT 2 : REAL, IMAGINARY ? 12,3
    COEFFICIENT 3 : REAL, IMAGINARY ? -2,2
    CONSTANT TERM 4 : REAL, IMAGINARY ? -4.33,-2.50
FOR EQUATION # 3
    COEFFICIENT 1 : REAL, IMAGINARY ? -5,0
    COEFFICIENT 2 : REAL, IMAGINARY ? -2,2
    COEFFICIENT 3 : REAL, IMAGINARY ? 17,-2
    CONSTANT TERM 4 : REAL, IMAGINARY ? 0,-10


THE SOLUTIONS ARE:

           REAL         IMAGINARY       MAGNITUDE      PHASE
X( 1 )     .866082      -.641543        1.07781        -36.93
X( 2 )     .0611964      .172112        .182668         71.2
X( 3 )     .230448       .439707        .496436         63.03

READY
>_
```

Fig. 5-33. Computed results for loop currents.

6

Active Filter Design

This chapter describes programs for the rapid design of various types of active filters. Those included are the state-variable filter (3 op-amps); second- through sixth-order low-pass and high-pass filters having either Bessel (best delay), Butterworth (maximally flat), or Chebyshev (equal ripple) responses; bandpass filters; and notch filters. Each design determines the performance based on user selected standard resistor and capacitor values. For a more detailed discussion about active filters, either the *Design of Active Filters, with Experiments,* or the *Active Filter Cookbook* (Howard W. Sams & Co., Inc., Indianapolis, IN 46206) are recommended references.

LOW-PASS AND HIGH-PASS FILTERS

The LPHP program listing shown in Fig. 6-1 enables us to design second- through sixth-order low-pass or high-pass filters having either Bessel, Butterworth, or Chebyshev (1-, 2-, or 3-dB) responses. Shown in Figs. 6-2 through 6-11 are the circuits of the low-pass and high-pass filters. Each second-order section is of the voltage-controlled voltage-source design (often referred to as a Sallen-Key filter) with both the frequency determining resistors R and capacitors C made equal. For the third- and fifth-order filters, the first section is a simple first-order passive network followed by a voltage follower. The damping values for each second-order section are controlled by resistors RA and RB. Because the program restricts the design of the filter to having both the frequency determining resistors and capacitors equal for each second-order section, the overall passband gain will be fixed, being set by the values chosen for RA and RB.

Example 6-1

Design a third-order, 350-Hz high-pass filter having a 1-dB Chebyshev response. When executed, the LPHP program requires the following initial information:

1. Type of filter
2. Order of filter
3. Low-pass or high-pass response
4. Cutoff frequency
5. Standard capacitor value

From this information, the remaining ideal component values are determined, after which we are asked to enter standard component values that are closest to the ideal values. The output results and final circuit for the design are shown in Figs. 6-12 and 6-13.

Example 6-2

Design a sixth-order, 3000-Hz Butterworth low-pass filter. As shown in the output results in Fig. 6-14, we only have to determine the frequency determining resistors and capacitors as for the first section, since they are the same for the remaining sections in a Butterworth design. The final circuit is shown in Fig. 6-15.

THE STATE-VARIABLE FILTER

The state-variable filter, shown in Fig. 6-16, uses three operational amplifiers, so that there are the following output responses: second-order low-pass, second-order high-pass, and a single pole bandpass. Resistors R and capacitors C determine the cutoff (or center) frequency, while resistors RA and RB control the damping (or Q).

```
100 'LOW AND HIGH-PASS FILTER DESIGN (LPHP)
101 CLS:CLEAR:DIM F$(5),VF(120)
102 GOTO 340
103 FOR V=1 TO 110
104 READ VF(V):NEXT V
105 CLS:PRINTF$(TY);" FILTER DESIGN"
106 PRINT:PRINT:PRINT"SELECT:":PRINT
107 INPUT"LOW-PASS (1) OR HIGH-PASS (2) ";FR
108 INPUT"ORDER OF FILTER ";O
109 IF O<2 OR O>6 GOTO 108  ELSE 110
110 INPUT"CUTOFF FREQUENCY IN HZ ";FC
111 ON O-1 GOTO 113  ,128  ,153  ,184  ,224
112 '2ND ORDER DESIGN
113 CLS:PRINT"1ST SECTION (2ND ORDER):"
114 PRINT:INPUT"C IN UF ";C
115 K=VF(((TY-1)*22)+1)
116 IF FR=1 THEN 118  ELSE 117
117 F=FC/K:GOTO 119
118 F=FC*K
119 R=1000/(6.28*F*C):PRINT"   THEN R =";INT(R*100+.5)/100;"K-OHMS"
120 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
121 IF Z$="YES" THEN 114  ELSE 122
122 INPUT"RA IN K-OHMS ";RA
123 D=VF((TY-1)*22+2)
124 RB=RA*(2-D):PRINT"   RB =";INT(RB*100+.5)/100;"K-OHMS"
125 INPUT"ANOTHER VALUE FOR RA (YES/NO) ";Z$
126 IF Z$="YES" THEN 122  ELSE 270
127 '3RD ORDER DESIGN
128 CLS:PRINT"1ST SECTION (1ST ORDER):"
129 PRINT:INPUT"C IN UF ";C1
130 K=VF(((TY-1)*22)+3)
131 IF FR=1 THEN 133  ELSE 132
132 F1=FC/K:GOTO 134
133 F1=FC*K
134 R1=1000/(6.28*F1*C1):PRINT"   THEN R =";INT(R1*100+.5)/100;"K-OHMS"
135 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
136 IF Z$="YES" THEN 129  ELSE 137
137 CLS:PRINT"2ND SECTION (2ND ORDER):"
138 IF TY=2 THEN 139  ELSE 140
139 C2=C1:R2=R1:F2=F1:PRINT:GOTO 148
140 PRINT:INPUT"C IN UF ";C2
141 K2=VF(((TY-1)*22)+5)
142 IF FR=1 THEN 144  ELSE 143
143 F2=FC/K2:GOTO 145
144 F2=FC*K2
145 R2=1000/(6.28*F2*C2):PRINT"   THEN R =";INT(R2*100+.5)/100;"K-OHMS"
146 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
147 IF Z$="YES" THEN 140  ELSE 148
148 D=VF((TY-1)*22+6)
149 INPUT"RA IN K-OHMS ";R3:R4=R3*(2-D):PRINT"   THEN RB =";INT(R4*100+.5)/100;"K-OHMS"
150 INPUT"TRY ANOTHER VALUE FOR RA (YES/NO) ";Z$
151 IF Z$="YES" THEN 149  ELSE 283
152 '4TH ORDER DESIGN
153 CLS:PRINT"1ST SECTION (2ND ORDER):"
154 K=VF(((TY-1)*22)+7)
155 PRINT:INPUT"C IN UF ";C1
156 IF FR=1 THEN 157  ELSE 158
157 F1=FC/K:GOTO 159
158 F1=FC*K
159 R1=1000/(6.28*F1*C1):PRINT"   THEN R =";INT(R1*100+.5)/100;"K-OHMS"
160 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
```

Fig. 6-1. Listing

```
161 IF Z$="YES" THEN 155  ELSE 162
162 D=VF((TY-1)*22+8)
163 INPUT"RA IN K-OHMS ";R3:R4=R3*(2-D)
164 PRINT"   THEN RB =";INT(R4*100+.5)/100;"K-OHMS"
165 INPUT"TRY ANOTHER VALUE FOR RA (YES/NO) ";Z$
166 IF Z$="YES" THEN 163  ELSE 167
167 CLS:PRINT"2ND SECTION (2ND ORDER):"
168 IF TY=2 THEN 169  ELSE 170
169 C2=C1:R2=R1:F2=F1:PRINT:GOTO 178
170 PRINT:INPUT"C IN UF ";C2
171 K2=VF(((TY-1)*22)+9)
172 IF FR=1 THEN 174  ELSE 173
173 F2=FC/K2:GOTO175
174 F2=FC*K2
175 R2=1000/(6.28*F2*C2):PRINT"   THEN R =";INT(R2*100+.5)/100;"K-OHMS"
176 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
177 IF Z$="YES" THEN 170  ELSE 178
178 D2=VF((TY-1)*22+10)
179 INPUT"RA IN K-OHMS ";R5:R6=R5*(2-D2)
180 PRINT"   THEN RB =";INT(R6*100+.5)/100;"K-OHMS"
181 INPUT"TRY ANOTHER VALUE FOR RA (YES/NO) ";Z$
182 IF Z$="YES" THEN 179  ELSE 297
183 '5TH ORDER DESIGN
184 CLS:PRINT"1ST SECTION (1ST ORDER):"
185 K=VF(((TY-1)*22)+11)
186 PRINT:INPUT"C IN UF ";C1
187 IF FR=1 THEN 189  ELSE 188
188 F1=FC/K:GOTO190
189 F1=FC*K
190 R1=1000/(6.28*F1*C1):PRINT"   THEN R =";INT(R1*100+.5)/100;"K-OHMS"
191 PRINT:INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
192 IF Z$="YES" THEN 186  ELSE 193
193 CLS:PRINT"2ND SECTION (2ND ORDER):
194 IF TY=2 THEN 195  ELSE 196
195 R2=R1:C2=C1:F2=F1:PRINT:GOTO 204
196 PRINT:INPUT"C IN UF ";C2
197 K2=(((TY-1)*22)+13)
198 IF FR=1 THEN 200  ELSE 199
199 F2=FC/K2:GOTO 201
200 F2=FC*K2
201 R2=1000/(6.28*F2*C2):PRINT"   THEN R =";INT(R2*100+.5)/100;"K-OHMS"
202 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
203 IF Z$="YES" THEN 196  ELSE 204
204 D2=VF((TY-1)*22+14)
205 INPUT"RA IN K-OHMS ";R4:R6=R4*(2-D2):PRINT"   THEN RB =";INT(R6*100+.5)/100;"K-OHMS"
206 INPUT"TRY ANOTHER VALUE FOR RA (YES/NO) ";Z$
207 IF Z$="YES" THEN 205  ELSE 208
208 CLS:PRINT"3RD SECTION (2ND ORDER):"
209 IF TY=2 THEN 210  ELSE 211
210 R3=R1:C3=C1:F3=F1:PRINT:GOTO 219
211 PRINT:INPUT"C IN UF ";C3
212 K3=(((TY-1)*22)+15)
213 IF FR=1 THEN 215  ELSE 214
214 F3=FC/K3:GOTO 216
215 F3=FC*K3
216 R3=1000/(6.28*F3*C3):PRINT"   THEN R =";INT(R3*100+.5)/100;"K-OHMS"
217 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
218 IF Z$="YES" THEN 211  ELSE 219
219 D3=VF((TY-1)*22+16)
220 INPUT"RA IN K-OHMS ";R5:R7=R5*(2-D3):PRINT"   THEN RB =";INT(R7*100+.5)/100;"K-OHMS"
221 INPUT"TRY ANOTHER VALUE FOR RA (YES/NO) ";Z$
222 IF Z$="YES" THEN 220  ELSE 311
```

Continued on next page.

for LPHP program.

```
223 '6TH ORDER DESIGN
224 CLS:PRINT"1ST SECTION (2ND ORDER):"
225 K=VF(((TY-1)*22)+17)
226 PRINT:INPUT"C IN UF ";C1
227 IF FR=1 THEN 229  ELSE 228
228 F1=FC/K:GOTO 230
229 F1=FC*K
230 R1=1000/(6.28*F1*C1):PRINT"   THEN R =";INT(R1*100+.5)/100;"K-OHMS"
231 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
232 IF Z$="YES" THEN 226  ELSE 233
233 D=VF((TY-1)*22+18)
234 INPUT"RA IN K-OHMS ";R4:R7=R4*(2-D)
235 PRINT"   THEN RB =";INT(R7*100+.5)/100;"K-OHMS"
236 INPUT"TRY ANOTHER VALUE FOR RA (YES/NO) ";Z$
237 IF Z$="YES" THEN 234  ELSE 238
238 CLS:PRINT"2ND SECTION (2ND ORDER):"
239 IF TY=2 THEN 240  ELSE 241
240 R2=R1:C2=C1:F2=F1:PRINT:GOTO 249
241 PRINT:INPUT"C IN UF ";C2
242 K2=VF(((TY-1)*22)+19)
243 IF FR=1 THEN 245  ELSE 244
244 F2=FC/K2:GOTO 246
245 F2=FC*K2
246 R2=1000/(6.28*F2*C2):PRINT"   THEN R =";INT(R2*100+.5)/100;"K-OHMS"
247 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
248 IF Z$="YES" THEN 241  ELSE 249
249 D2=VF((TY-1)*22+20)
250 INPUT"RA IN K-OHMS ";R5:R8=R5*(2-D2)
251 PRINT"   THEN RB =";INT(R8*100+.5)/100;"K-OHMS"
252 INPUT"TRY ANOTHER VALUE FOR RA (YES/NO) ";Z$
253 IF Z$="YES" THEN 250  ELSE 254
254 CLS:PRINT"3RD SECTION (2ND ORDER):"
255 IF TY=2 THEN 256  ELSE 257
256 F3=F1:R3=R1:C3=C1:PRINT:GOTO 265
257 PRINT:INPUT"C IN UF ";C3
258 K3=VF(((TY-1)*22)+21)
259 IF FR=1 THEN 261  ELSE 260
260 F3=FC/K3:GOTO 262
261 F3=FC*K3
262 R3=1000/(6.28*F3*C3):PRINT"   THEN R =";INT(R3*100+.5)/100;"K-OHMS"
263 INPUT"TRY ANOTHER VALUE FOR C (YES/NO) ";Z$
264 IF Z$="YES" THEN 257  ELSE 265
265 D3=VF((TY-1)*22+22)
266 INPUT"RA IN K-OHMS ";R6:R9=R6*(2-D3)
267 PRINT"   THEN RB =";INT(R9*100+.5)/100;"K-OHMS"
268 INPUT"TRY ANOTHER VALUE FOR RA (YES/NO) ";Z$
269 IF Z$="YES" THEN 266  ELSE 325
270 CLS:PRINT"DESIGN SUMMARY: 2ND ORDER ";F$(TY);" FILTER"
271 PRINT"C (UF)",C
272 PRINT"R (K-OHMS)",INT(R*100+.5)/100
273 PRINT"RA (K-OHMS)",INT(RA*100+.5)/100
274 PRINT"RB (K-OHMS)",INT(RB*100+.5)/100
275 PRINT:PRINT"SELECT STANDARD VALUES FOR:"
276 INPUT"C (IN UF) ";C
277 INPUT"R (IN K-OHMS) ";R
278 INPUT"RA (IN K-OHMS) ";RA
279 INPUT"RB (IN K-OHMS) ";RB
280 G=1+(RB/RA):DB=20*LOG(G)/LOG(10)
281 PRINT"PASSBAND GAIN FIXED AT ";INT(DB*100+.5)/100;"DB"
282 GOTO 363
```

Fig. 6-1 (cont). Listing

```
283 CLS:PRINT"DESIGN SUMMARY: 3RD ORDER ";F$(TY);" FILTER"
284 PRINT:PRINT" ","1ST SECTION","2ND SECTION"
285 PRINT"C (UF)",C1,C2
286 PRINT"R (K-OHMS)",INT(R1*100+.5)/100,INT(R2*100+.5)/100
287 PRINT"RA (K-OHMS)",,INT(R3*100+.5)/100
288 PRINT"RB (K-OHMS)",,INT(R4*100+.5)/100
289 PRINT:PRINT"INPUT STANDARD VALUES FOR:"
290 INPUT"C (IN UF)";C1,C2
291 INPUT"R (IN K-OHMS)";R1,R2
292 INPUT"RA (IN K-OHMS)";R3
293 INPUT"RB (IN K-OHMS)";R4
294 G=1+(R4/R3):DB=20*LOG(G)/LOG(10)
295 PRINT"PASSBAND GAIN FIXED AT ";INT(DB*100+.5)/100;"DB"
296 GOTO 363
297 CLS:PRINT"DESIGN SUMMARY: 4TH ORDER ";F$(TY);" FILTER"
298 PRINT:PRINT" ","1ST SECTION","2ND SECTION"
299 PRINT"C (UF) ",C1,C2
300 PRINT"R (K-OHMS) ",INT(R1*100+.5)/100,INT(R2*100+.5)/100
301 PRINT"RA (K-OHMS) ",INT(R3*100+.5)/100,INT(R5*100+.5)/100
302 PRINT"RB (K-OHMS) ",INT(R4*100+.5)/100,INT(R6*100+.5)/100
303 PRINT:PRINT"SELECT STANDARD VALUES FOR:"
304 INPUT"C (IN UF) ";C1,C2
305 INPUT"R (IN K-OHMS) ";R1,R2
306 INPUT"RA (IN K-OHMS) ";R3,R4
307 INPUT"RB (IN K-OHMS) ";R5,R6
308 G=(1+(R5/R3))*(1+(R6/R4)):DB=20*LOG(G)/LOG(10)
309 PRINT"PASSBAND GAIN FIXED AT ";INT(DB*100+.5)/100;"DB"
310 GOTO 363
311 CLS:PRINT"DESIGN SUMMARY: 5TH ORDER ";F$(TY);" FILTER"
312 PRINT:PRINT" ","1ST SECTION","2ND SECTION","3RD SECTION"
313 PRINT"C (UF)",C1,C2,C3
314 PRINT"R (K-OHMS)",INT(R1*100+.5)/100,INT(R2*100+.5)/100,INT(R3*100+.5)/100
315 PRINT"RA (K-OHMS)",,INT(R4*100+.5)/100,INT(R5*100+.5)/100
316 PRINT"RB (K-OHMS)",,INT(R6*100+.5)/100,INT(R7*100+.5)/100
317 PRINT:PRINT"SELECT STANDARD VALUES FOR:"
318 INPUT"C (IN UF) ";C1,C2,C3
319 INPUT"R (IN K-OHMS) ";R1,R2,R3
320 INPUT"RA (IN K-OHMS) ";R4,R5
321 INPUT"RB (IN K-OHMS) ";R6,R7
322 G=(1+(R6/R4))*(1+(R7/R5)):DB=20*LOG(G)/LOG(10)
323 PRINT"PASSBAND GAIN FIXED AT ";INT(DB*100+.5)/100;"DB"
324 GOTO 363
325 CLS:PRINT"DESIGN SUMMARY: 6TH ORDER ";F$(TY);" FILTER"
326 PRINT:PRINT" ","1ST SECTION","2ND SECTION","3RD SECTION"
327 PRINT"C (UF)",C1,C2,C3
328 PRINT"R (K-OHMS)",INT(R1*100+.5)/100,INT(R2*100+.5)/100,INT(R3*100+.5)/100
329 PRINT"RA (K-OHMS)",INT(R4*100+.5)/100,INT(R5*100+.5)/100,INT(R6*100+.5)/100
330 PRINT"RB (K-OHMS)",INT(R7*100+.5)/100,INT(R8*100+.5)/100,INT(R9*100+.5)/100
331 PRINT:PRINT"SELECT STANDARD VALUES FOR:"
332 INPUT"C (IN UF) ";C1,C2,C3
333 INPUT"R (IN K-OHMS) ";R1,R2,R3
334 INPUT"RA (IN K-OHMS) ";R4,R5,R6
335 INPUT"RB (IN K-OHMS) ";R7,R8,R9
336 G=(1+(R7/R4))*(1+(R8/R5))*(1+(R9/R6)):DB=20*LOG(G)/LOG(10)
337 PRINT"PASSBAND GAIN FIXED AT ";INT(DB*100+.5)/100;"DB"
338 GOTO 363
339 'MENU SELECTION
340 CLS:PRINT,"LOW-PASS AND HIGH-PASS FILTER DESIGN"
341 PRINT:PRINT"SELECT TYPE OF FILTER RESPONSE:"
342 PRINT"      1. BESSEL"
343 PRINT"      2. BUTTERWORTH"
```

Continued on next page.

for LPHP program.

```
344 PRINT"     3. 1-DB CHEBYSHEV"
345 PRINT"     4. 2-DB CHEBYSHEV"
346 PRINT"     5. 3-DB CHEBYSHEV"
347 PRINT:INPUTTY
348 IF TY>5 THEN 340
349 F$(1)="BESSEL":F$(2)="BUTTERWORTH":F$(3)="1-DB CHEBYSHEV"
350 F$(4)="2-DB CHEBYSHEV":F$(5)="3-DB CHEBYSHEV"
351 GOTO 103
352 'DESIGN PARAMETERS
353 DATA 1.274,1.732,1.328,0,1.454,1.447,1.436,1.916,1.610,1.241,1.557,0
354 DATA 1.613,1.775,1.819,1.091,1.609,1.959,1.694,1.636,1.910,0.977
355 DATA 1,1.414,1,0,1,1,1,1.848,1,0.765,1,0
356 DATA 1,1.618,1,0.618,1,1.932,1,1.414,1,0.518
357 DATA 0.863,1.045,0.452,0,0.911,0.496,0.502,1.275,0.943,0.281,0.280,0
358 DATA 0.634,0.714,0.961,0.181,0.347,1.314,0.733,0.455,0.977,0.125
359 DATA 0.852,0.895,0.322,0,0.913,0.412,0.466,1.088,0.946,0.234,0.223,0
360 DATA 0.624,0.578,0.964,0.142,0.312,1.121,0.727,0.363,0.976,0.0989
361 DATA 0.841,0.767,0.299,0,0.916,0.326,0.443,0.929,0.950,0.179,0.178,0
362 DATA 0.614,0.468,0.967,0.113,0.298,0.958,0.722,0.289,0.975,0.0782
363 END
```

Fig. 6-1 (cont). Listing for LPHP program.

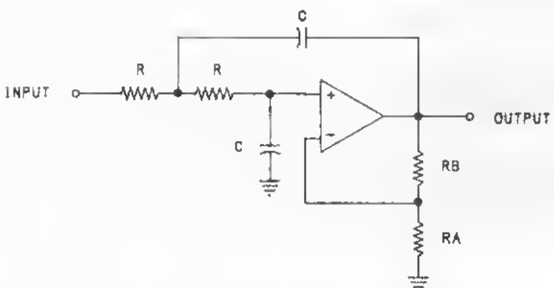


Fig. 6-2. Second-order low-pass filter.

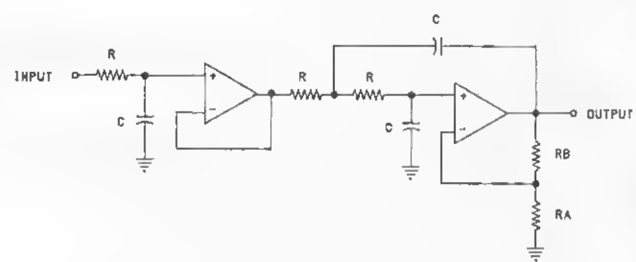


Fig. 6-4. Third-order low-pass filter.

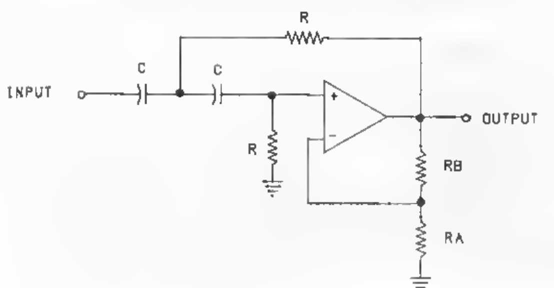


Fig. 6-3. Second-order high-pass filter.

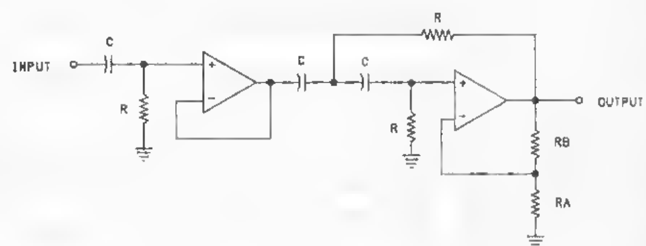


Fig. 6-5. Third-order high-pass filter.

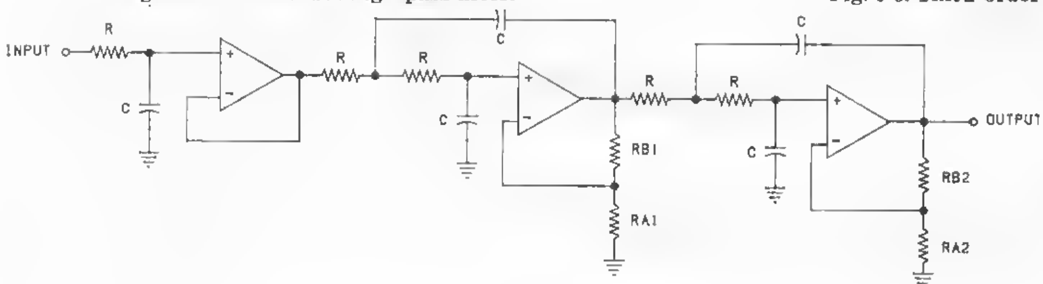


Fig. 6-6. Fourth-order low-pass filter.

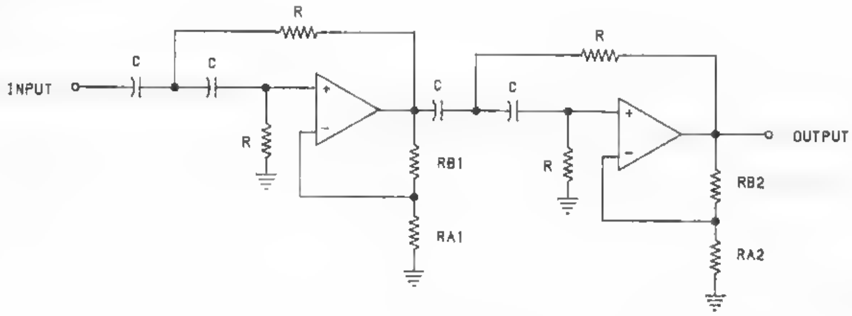


Fig. 6-7. Fourth-order high-pass filter.

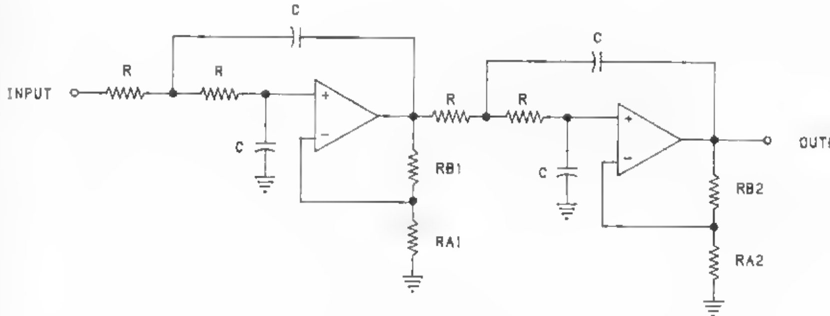


Fig. 6-8. Fifth-order low-pass filter.

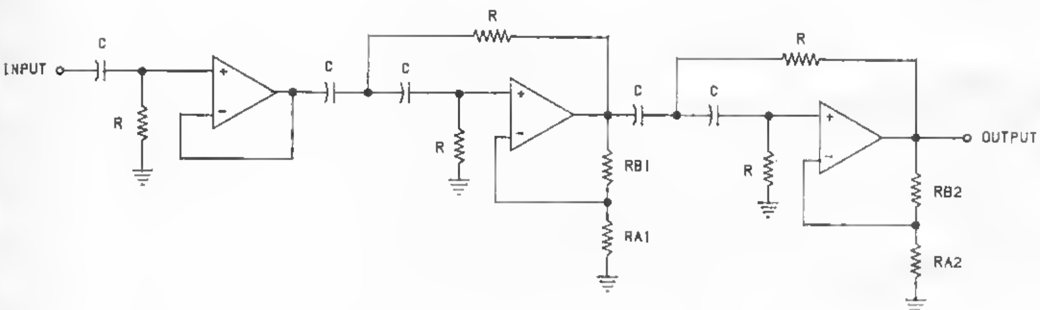


Fig. 6-9. Fifth-order high-pass filter.

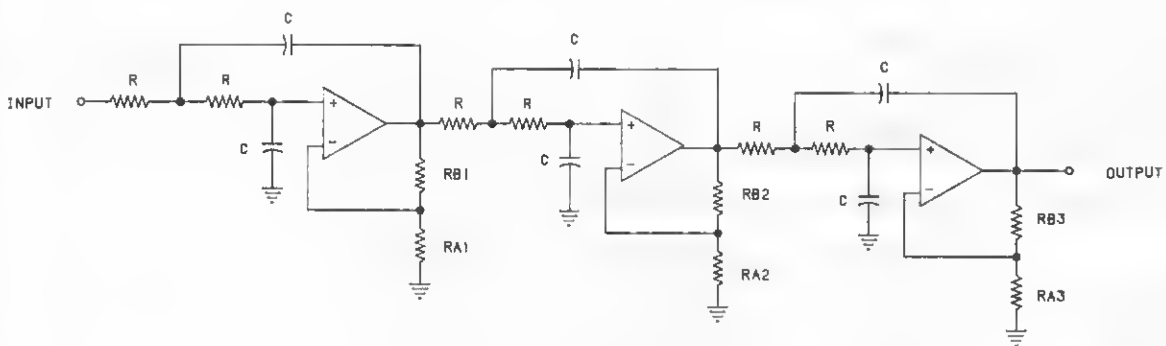


Fig. 6-10. Sixth-order low-pass filter.

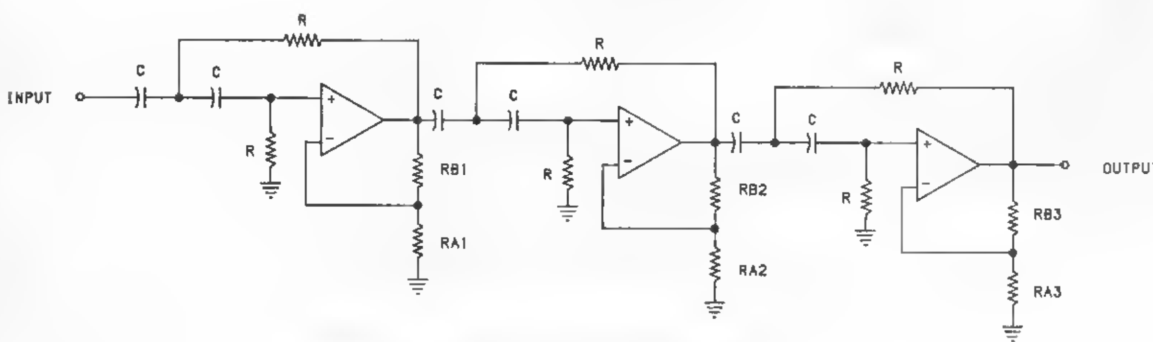


Fig. 6-11. Sixth-order high-pass filter.

```
                    LOW-PASS AND HIGH-PASS FILTER DESIGN

SELECT TYPE DF FILTER RESPDNSE:
     1. BESSEL
     2. BUTTERWORTH
     3. 1-DB CHEBYSHEV
     4. 2-DB CHEBYSHEV
     5. 3-DB CHEBYSHEV

? 3
1-DB CHEBYSHEV FILTER DESIGN


SELECT:

LOW-PASS (1) DR HIGH-PASS (2) ? 2
DRDER DF FILTER ? 3
CUTOFF FREQUENCY IN HZ ? 350

1ST SECTION (1ST ORDER):

C IN UF ? .01
   THEN R = 20.56 K-DHMS
TRY ANDTHER VALUE FDR C (YES/ND) ? ND

2ND SECTIDN (2ND ORDER):

C IN UF ? .022
   THEN R = 18.84 K-OHMS
TRY ANDTHER VALUE FOR C (YES/NO) ? YES

C IN UF ? .033
   THEN R = 12.56 K-OHMS
TRY ANDTHER VALUE FOR C (YES/ND) ? NO
RA IN K-OHMS ? 10
   THEN RB = 15.04 K-OHMS
TRY ANOTHER VALUE FOR RA (YES/ND) ? ND

DESIGN SUMMARY: 3RD ORDER 1-DB CHEBYSHEV FILTER

                 1ST SECTION     2ND SECTION
C (UF)            .01             .033
R (K-DHMS)        20.56           12.56
RA (K-DHMS)                       10
RB (K-OHMS)                       15.04

INPUT STANDARD VALUES FOR:
C (IN UF)? .01,.033
R (IN K-OHMS)? 20.5,12.7
RA (IN K-OHMS)? 10
RB (IN K-OHMS)? 15
PASSBAND GAIN FIXED AT  7.96 DB

READY
>_
```

Fig. 6-12. Output results for Example 6-1.

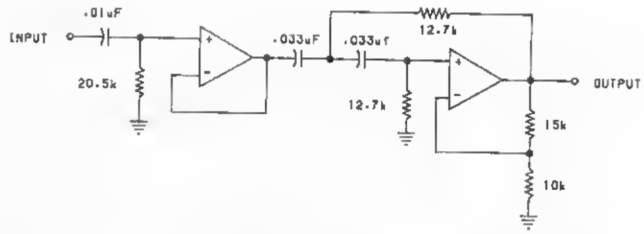


Fig. 6-13. Final circuit for Example 6-1.

Fig. 6-14. Results for Example 6-2.

```
                    LOW-PASS AND HIGH-PASS FILTER DESIGN
SELECT TYPE OF FILTER RESPONSE:
     1. BESSEL
     2. BUTTERWORTH
     3. 1-DB CHEBYSHEV
     4. 2-DB CHEBYSHEV
     5. 3-DB CHEBYSHEV

? 2
BUTTERWORTH FILTER DESIGN

SELECT:

LOW-PASS (1) OR HIGH-PASS (2) ? 1
ORDER OF FILTER ? 6
CUTOFF FREQUENCY IN HZ ? 3000

1ST SECTION (2ND ORDER):

C IN UF ? .0022
   THEN R = 24.13 K-OHMS
TRY ANOTHER VALUE FOR C (YES/NO) ? NO
RA IN K-OHMS ? 100
   THEN RB = 6.8 K-OHMS
TRY ANOTHER VALUE FOR RA (YES/NO) ? NO

2ND SECTION (2ND ORDER):

RA IN K-OHMS ? 18
   THEN RB = 10.55 K-OHMS
TRY ANOTHER VALUE FOR RA (YES/NO) ? NO

3RD SECTION (@2ND ORDER):

RA IN K-OHMS ? 12
   THEN RB = 17.78 K-OHMS
TRY ANOTHER VALUE FOR RA (YES/NO) ? NO

DESIGN SUMMARY: 6TH ORDER BUTTERWORTH FILTER

                  1ST SECTION     2ND SECTION     3RD SECTION
C (UF)             2.2E-03         2.2E-03         2.2E-03
R (K-OHMS)         24.13           24.13           24.13
RA (K-OHMS)        100             18              12
RB (K-OHMS)        6.8             10.55           17.78

SELECT STANDARD VALUES FOR:
C (IN UF) ? .0022,.0022,.0022
R (IN K-OHMS) ? 24,24,24
RA (IN K-OHMS) ? 100,18,12
RB (IN K-OHMS) ? 6.8,10.5,17.8
PASSBAND GAIN FIXED AT  12.46 DB

READY
>_
```

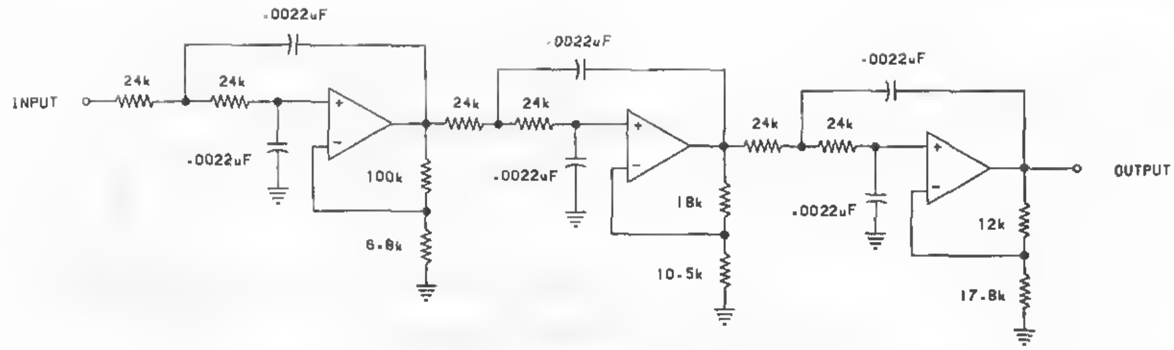


Fig. 6-15. Final circuit for Example 6-2.

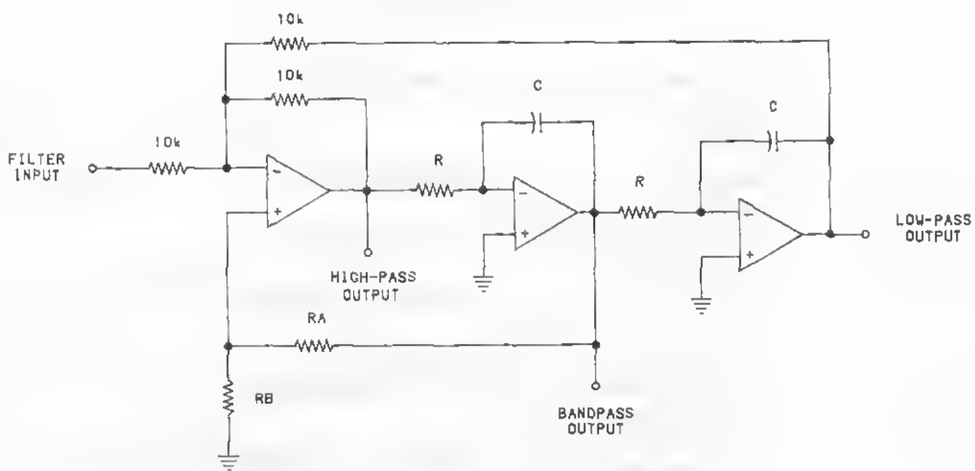


Fig. 6-16. Basic state-variable filter.

For the low-pass and high-pass outputs, the passband gain is unity. For the bandpass output, the center frequency gain will be equal to the value picked for the Q of the filter.

As this type of circuit gives a second-order response for both the low-pass and high-pass outputs, it will not be possible to obtain optimum performance with all three outputs simultaneously. For either a low-pass or a high-pass Butterworth response, Q must be equal to 0.707 (damping equals 1.414, or 1/Q). Consequently, the bandpass response suffers terribly! Even for a second-order 3-dB Chebyshev filter, Q must be 1.3, which is not much better. We should then design either for a second-order Butterworth low-pass/high-pass response (Q = 0.707), or for a high-Q bandpass response.

An additional feature of the basic state-variable filter is an additional operational amplifier stage, as shown in Fig. 6-17, to simultaneously add the low-pass and high-pass filter outputs to create a notch or band reject filter.

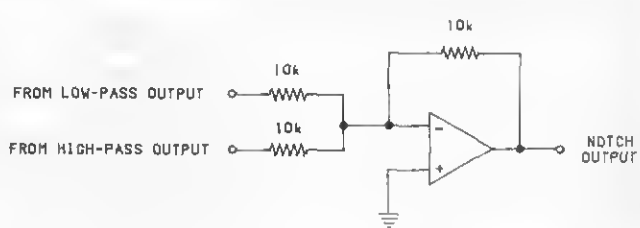


Fig. 6-17. Creating a notch filter with an additional op-amp summing amplifier.

Shown in Fig. 6-18 is the program listing (SV-FILTER) for designing state-variable filters, which requires the following initial information:

1. Type of response (low, high, bandpass, or notch)
2. Cutoff, center, or notch frequency
3. Damping factor, or Q
4. Standard capacitor value

From this information, the remaining component values are determined, after which we are asked to enter standard component values that are clos-

```
100 'STATE-VARIABLE FILTER DESIGN (SVFILTER)
101 CLEAR:CLS:PRINTTAB(18)"STATE-VARIABLE FILTER DESIGN"
102 PRINT
103 PRINT"OUTPUT RESPONSES AVAILABLE:"
104 PRINT"      1. 2ND ORDER LOW-PASS"
105 PRINT"      2. 2ND ORDER HIGH-PASS"
106 PRINT"      3. 1-POLE BANDPASS"
107 PRINT"      4. 1-POLE NOTCH"
108 PRINT@478,"SELECT # ":Z9=490:GOSUB237
109  IF Z9$<>"1" AND Z9$<>"2" AND Z9$<>"3" AND Z9$<>"4" THEN 108  ELSE 110
110 IF Z9$="1" THEN 114
111 IF Z9$="2" THEN 149
112 IF Z9$="3" THEN 184
113 IF Z9$="4" THEN 184
114 CLS:PRINT:PRINT"2ND ORDER LOW-PASS STATE-VARIABLE FILTER"
115 PRINT:INPUT"ENTER CUTOFF FREQUENCY IN HZ "; FC
116 INPUT"ENTER DAMPING FACTOR ";D
117 IF D>=3 THEN 118  ELSE 119
118 PRINT"DAMPING FACTOR TOO LARGE (>3). PICK ANOTHER VALUE":GOTO 116
119 PRINT:PRINT"PASSBAND GAIN FIXED AT 1.0 (0 DB)"
120 PRINT:INPUT"ENTER CAPACITOR VALUE (IN UF) ";C:P=6.28319*FC
121 R=1000/(P*C):E=(3/D)-1
122 PRINT"      THEN R = ";INT(R*100+.5)/100;" K-OHMS"
123 INPUT"ENTER VALUE FOR RA ";RA
124 RB=RA/E
125 PRINT"      THEN RB =";INT(RB*100+.5)/100;"K-OHMS"
126 PRINT:INPUT"TRY ANOTHER SET OF VALUES(YES/NO) ";A$
127 IF A$="YES" THEN 120  ELSE 128
128 PRINT:PRINT"ENTER STANDARD COMPONENT VALUES FOR:"
129 PRINT:INPUT"C (IN UF) ";C
130 INPUT"R (IN K-OHMS) ";R
131 INPUT"RA (IN K-OHMS) ";RA
132 INPUT"RB (IN K-OHMS) ";RB
133 PRINT:PRINT"BASED ON ABOVE COMPONENTS CHOSEN:"
134 FC=1000/(6.28319*R*C):D=(3*RB)/(RA+RB)
135 PRINT"CUTOFF FREQUENCY = ";INT(FC*10+.5)/10;"HZ"
136 PRINT"DAMPING FACTOR = ";INT(D*1000+.5)/1000
137 PRINT:INPUT"TRY ANOTHER SET OF STANDARD VALUES (YES/NO) ";A$
138 IF A$="YES" THEN 128  ELSE 139
139 CLS:INPUT"AMPLITUDE RESPONSE AT A GIVEN FREQUENCY (YES/NO) ";A$
140 IF A$="YES" THEN 141  ELSE 148
141 INPUT"FREQUENCY IN HZ ";FI
142 F=FI/FC
143 DB=-20*LOG((F↑4)+((D↑2-2)*F↑2)+1)/LOG(10)
144 DB=DB/2
145 PRINT"    AMPLITUDE RESPONSE = ";INT(DB*100+.5)/100;" DB"
146 PRINT:PRINT"TRY ANOTHER FREQUENCY (YES/NO) ";A$
147 IF A$="YES" THEN 141  ELSE 148
148 PRINT"DESIGN COMPLETED":END
149 CLS:PRINT:PRINT"2ND ORDER HIGH-PASS STATE-VARIABLE FILTER"
150 PRINT:INPUT"ENTER CUTOFF FREQUENCY IN HZ ";FC
151 INPUT"ENTER DAMPING FACTOR ";D
152 IF D>=3 THEN 153  ELSE 154
153 PRINT"DAMPING FACTOR TOO LARGE (>3). PICK ANOTHER VALUE":GOTO 151
154 PRINT:PRINT"PASSBAND GAIN FIXED AT 1.0 (0 DB)"
155 PRINT:INPUT"ENTER CAPACITOR VALUE IN UF ";C:P=6.28319*FC
156 R=1000/(P*C):E=(3/D)-1
157 PRINT"      THEN R = ";INT(R*100+.5)/100;" K-OHMS"
158 INPUT"ENTER VALUE FOR RA (IN K-OHMS) ";RA
159 RB=RA/E
160 PRINT"      THEN RB = ";INT(RB*100+.5)/100;"K-OHMS"
161 INPUT"TRY ANOTHER SET OF VALUES (YES/NO) ";A$
```

Continued on next page.

Fig. 6-18. Listing for SVFILTER program.

```
162 IF A$="YES" THEN 155  ELSE 163
163 PRINT:PRINT"ENTER STANDARD COMPONENT VALUES FOR:"
164 PRINT:INPUT"C (IN UF) ";C
165 INPUT"R (IN K-OHMS) ";R
166 INPUT"ra IN K-OHMS ";RA:INPUT"RB IN K-OHMS ";RB
167 PRINT:PRINT"BASED ON ABOVE COMPONENTS CHOSEN:"
168 FC=1000/(6.28319*R*C):D=(3*RB)/(RA+RB)
169 PRINT"CUTOFF FREQUENCY = ";INT(FC*10+.5)/10;"HZ"
170 PRINT"DAMPING FACTOR = ";INT(D*1000+.5)/1000
171 INPUT"TRY ANOTHER SET OF VALUES (YES/NO) ";A$
172 IF A$="YES" THEN 163  ELSE 173
173 CLS:INPUT"AMPLITUDE RESPONSE AT A GIVEN FREQUENCY (YES/NO)";A$
174 IF A$="YES" THEN 175  ELSE 183
175 INPUT"FREQUENCY IN HZ ";FI
176 F=FC/FI
177 DB=-20*LOG((F↑4)+((D↑2-2)*F↑2)+1)/LOG(10)
178 DB=DB/2
179 PRINT"    AMPLITUDE RESPONSE = ";INT(DB*100+.5)/100;" DB"
180 INPUT"ADDITIONAL FREQUENCIES (1=YES, 0=NO)";A
181 IF A<>1 AND A<>0 THEN 180  ELSE 182
182 IF A=1 THEN 175  ELSE 183
183 PRINT"DESIGN COMPLETED":END
184 IF Z9$="3" GOTO 185  ELSE 187
185 CLS:PRINT:PRINT"1-POLE BANDPASS STATE-VARIABLE FILTER"
186 PRINT:INPUT"ENTER CENTER FREQUENCY IN HZ ";FC:GOTO 189
187 CLS:PRINT:PRINT"1-POLE NOTCH STATE-VARIABLE FILTER"
188 PRINT:INPUT"ENTER NOTCH (NULL) FREQUENCY IN HZ ";FC
189 INPUT"3 DB BANDWIDTH IN HZ ";BW:Q=FC/BW
190 INPUT"ENTER CAPACITOR VALUE IN UF ";C
191 P=6.28319*FC
192 R=1000/(P*C)
193 E=(3*Q)-1
194 PRINT"    THEN R = ";INT(R*100+.5)/100;" K-OHMS"
195 INPUT"ENTER VALUE FOR RA (IN K-OHMS) ";RA
196 RB=RA/E
197 PRINT"     THEN RB = ";INT(RB*100+.5)/100;"K-OHMS"
198 INPUT"TRY ANOTHER SET OF VALUES (YES/NO) ";A$
199 IF A$="YES" THEN 190  ELSE 200
200 PRINT:PRINT"ENTER STANDARD COMPONENT VALUES FOR:"
201 INPUT"C (IN UF) ";C
202 INPUT"R (IN K-OHMS) ";R
203 E=(3*Q)-1
204 INPUT"RA IN K-OHMS ";RA:INPUT"RB IN K-OHMS";RB
205 IF Z9$="3"  THEN 206  ELSE 207
206 PRINT:PRINT"CENTER FREQUENCY GAIN FIXED AT Q (";Q;")":GOTO 208
207 PRINT"PASSBAND GAIN FIXED AT 1.0 (0 DB)"
208 FC=1000/(6.28319*R*C):Q=(1+(RA/RB))/3:BW=FC/Q
209 PRINT:PRINT"BASED ON ABOVE COMPONENT VALUES:"
210 IF T=3 THEN 211  ELSE 212
211 PRINT:PRINT"CENTER FREQUENCY = ";FC;" HZ":GOTO 213
212 PRINT:PRINT"NOTCH (NULL) FREQUENCY = ";FC;" HZ"
213 PRINT"Q = ";Q
214 PRINT"3 DB BANDWIDTH = ";BW;" HZ"
215 IF T=3 THEN 216  ELSE 218
216 DB=8.68589*LOG(Q):PRINT"CENTER FREQUENCY GAIN = ";DB;"DB"
217 GOTO 219
218 PRINT"PASSBAND GAIN = 0 DB"
219 INPUT"ANOTHER SET OF NOMINAL VALUES (1=YES, 0=NO)";A
220 IF A=1 THEN 200  ELSE 221
221 CLS:INPUT"AMPLITUDE RESPONSE AT A GIVEN FREQUENCY (1=YES, 0=NO)";A
222 IF A=1 THEN 223  ELSE 236
```

Continued on next page.

Fig. 6-18 (cont). Listing for SVFILTER program.

```
223 INPUT"INPUT FREQUENCY IN HZ ";FI
224 F=FI/FC
225 IF T=3 THEN 226  ELSE 231
226 DB=8.6859*LOG(Q)-4.3429*LOG(1+Q↑2*((F↑2-1)/F)↑2)
227 PRINT"     AMPLITUDE RESPONSE = ";DB;" DB"
228 INPUT"ADDITIONAL FREQUENCIES (1=YES, 0=NO) ";A
229 IF A<>0 AND A<>1 THEN 228  ELSE 230
230 IF A=1 THEN 223  ELSE 236
231 DB=4.34294*LOG((1-F↑2)↑2)-4.34294*LOG((1-F↑2)↑2+(F/Q)↑2)
232 PRINT"     AMPLITUDE RESPONSE = ";DB;" DB"
233 INPUT"ADDITIONAL FREQUENCIES (1=YES, 0=NO) ";A
234 IF A<>0 AND A<>1 THEN 233  ELSE 235
235 IF A=1 THEN 223  ELSE 236
236 PRINT"DESIGN COMPLETED":END
237 PRINT@Z9,CHR$(143);:L=1          'BLINKING CURSOR ROUTINE
238 Z9$=INKEY$
239 IFZ9$<>"" THEN PRINT @ Z9," ";:RETURN
240 L=L+1
241 IF L<10 THEN 238
242 PRINT @Z9," ";
243 FOR L=1 TO 5:NEXT
244 GOTO 237
```

Fig. 6-18 (cont). Listing for SVFILTER program.

est to the ideal values. In addition, we are then able to determine the response of the filter at a given frequency, based on the standard component values selected.

Example 6-3

Design a second-order high-pass state-variable filter having a cutoff frequency of 350 Hz with a Butterworth response (damping equals 1.414). Choosing a capacitor value of 0.022 μF, Fig. 6-19 shows the resulting output computations, with the final circuit shown in Fig. 6-20.

A BANDPASS FILTER FOR Q LESS THAN 10

For values of Q less than 10, the multiple-feedback circuit of Fig. 6-21 is often used, since it uses a minimum of components. Furthermore, the design is simplified by making both frequency determining capacitors equal. Resistors R1 and R3 determine the center frequency passband gain, while the center frequency is set by all three resistors. The program listing (BNDPASS1) is given in Fig. 6-22.

When executed the program requires the following initial information:

1. Design determined by knowing either Q, 3-dB bandwidth, or the 3-dB frequencies
2. Standard capacitor value
3. Center frequency
4. Center frequency voltage gain

after which the resistor values are determined. In addition, we are then able to determine the filter response at any frequency based on the standard values chosen. The following example illustrates the BNDPASS1 program.

Example 6-4

Design a bandpass filter with a center frequency of 750 Hz, a voltage gain of 1.3 (+2.3 dB), and a Q of 4.2. Using a standard capacitor value of 0.01 μF, the three resistor values are calculated after which we are asked whether or not we want to try a different set of values, particularly that of the capacitor. If not, we are asked to enter standard component values that are closest to those calculated by the program. Again, we are asked whether or not we want to change the values. If not, then we are asked if we want to know the frequency response of the filter at a particular frequency, based on the component values we have just chosen.

Figs. 6-23 and 6-24 show both the program output and the final circuit.

```
2ND ORDER HIGH-PASS STATE-VARIABLE FILTER

ENTER CUTOFF FREQUENCY IN HZ ? 350
ENTER DAMPING FACTOR ? 1.414

PASSBAND GAIN FIXED AT 1.0 (0 DB)

ENTER CAPACITOR VALUE IN UF ? 0.022
     THEN R =  20.67  K-OHMS
ENTER VALUE FOR RA (IN K-OHMS) ? 27
     THEN RB =  24.07 K-OHMS
TRY ANOTHER SET OF VALUES (YES/NO) ? NO


ENTER STANDARD COMPONENT VALUES FOR:

C (IN UF) ? 0.022
R (IN K-OHMS) ? 20
RA IN K-OHMS ? 27
RB IN K-OHMS ? 24

BASED ON ABOVE COMPONENTS CHOSEN:
CUTOFF FREQUENCY =  361.7 HZ
DAMPING FACTOR =  1.412
TRY ANOTHER SET OF VALUES (YES/NO) ? NO

AMPLITUDE RESPONSE AT A GIVEN FREQUENCY (YES/NO)? YES
FREQUENCY IN HZ ? 500
    AMPLITUDE RESPONSE = -1.04  DB
ADDITIONAL FREQUENCIES (1=YES, 0=NO)? 1
FREQUENCY IN HZ ? 250
    AMPLITUDE RESPONSE = -7.3  DB
ADDITIONAL FREQUENCIES (1=YES, 0=NO)? 1
FREQUENCY IN HZ ? 100
    AMPLITUDE RESPONSE = -22.36  DB
ADDITIONAL FREQUENCIES (1=YES, 0=NO)? 0

DESIGN COMPLETED


READY
>
```

Fig. 6-19. Results for Example 6-3.

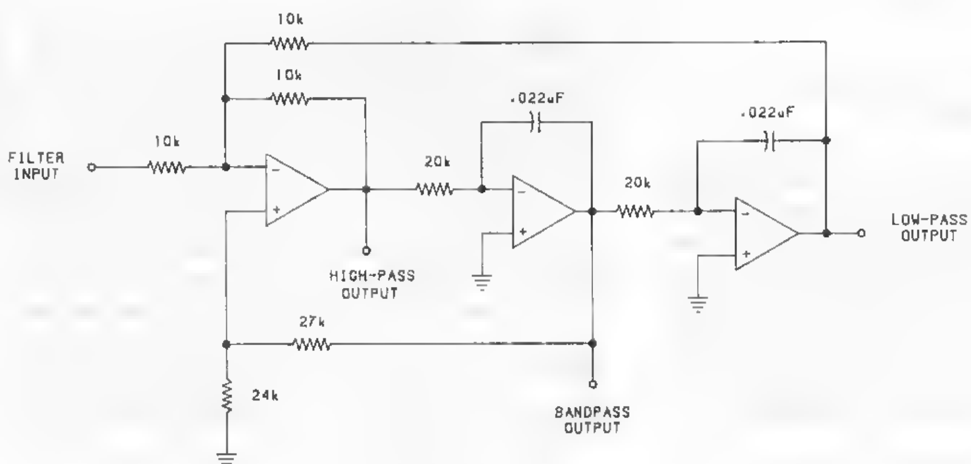


Fig. 6-20. Final circuit for Example 6-3.

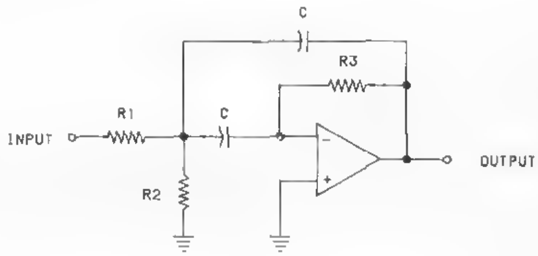


Fig. 6-21. Bandpass filter for Q less than 10.

```
100 ' BANDPASS FILTER DESIGN FOR Q < 10 (BNDPASS1)
101 CLEAR:CLS:PRINTTAB(12)"BANDPASS FILTER DESIGN FOR Q LESS THAN 10"
102 PRINT
103 G$="####.##"
104 PRINT"DESIGN BASED ON KNOWING: 1. Q"
105 PRINT"                         2. 3-DB BANDWIDTH"
106 PRINT"                         3. UPPER & LOWER 3-DB FREQUENCIES"
107 PRINT@473,"SELECT # ":Z9=483:GOSUB174
108 IF Z9$<>"1" AND Z9$<>"2" AND Z-$<>"3" THEN 107  ELSE 109
109 PRINT@483,Z9$:IF Z9$="1" THEN 112
110 IF Z9$="2" THEN 116
111 IF Z9$="3" THEN 118
112 PRINT:INPUT"Q ";Q
113 IF Q>=10 THEN 114  ELSE 115
114 PRINT"Q MUST BE LESS THAN 10.":GOTO 112
115 GOTO 128
116 PRINT:INPUT"3-DB BANDWIDTH IN HZ ";BW
117 GOTO 128
118 PRINT:INPUT"LOWER 3-DB FREQUENCY IN HZ ";FL
119 INPUT"UPPER 3-DB FREQUENCY IN HZ ";FH
120 FO=SQR(FH*FL)
121 PRINT"   CENTER FREQUENCY =";INT(FO*10+.5)/10;"HZ"
122 BW=FH-FL:Q=FO/BW
123 PRINT"   3-DB BANDWIDTH =";INT(BW*10+.5)/10;"HZ"
124 PRINT"   Q =";INT(Q*10+.5)/10
125 IF Q>=10 THEN 126  ELSE 127
126 PRINT"Q MUST BE LESS THAN 10. INCREASE BANDWIDTH.":PRINT:GOTO 118
127 GOTO 128
128 PRINT:INPUT"HIT <ENTER> TO CONTINUE ";Z
129 CLS:INPUT"STANDARD CAPACITOR VALUE IN UF ";C
130 IF Z9$="1" THEN 133  ELSE 131
131 IF Z9$="3" THEN 136  ELSE 132
132 IF Z9$="2" THEN 133
133 INPUT"CENTER FREQENCY IN HZ ";FO
134 IF Z9$="1" THEN 136  ELSE 135
135 Q=FO/BW
136 INPUT"CENTER FREQUENCY VOLTAGE GAIN ";G
137 IF Q>=SQR(G/2) THEN 140
138 PRINT"NO SOLUTION POSSIBLE FOR STATED Q AND GAIN - TRY A"
139 PRINT"LOWER CENTER FREQUENCY GAIN":GOTO 136
140 R1=1000*Q/(G*6.28*FO*C):R3=2000*Q/(6.28*FO*C)
141 R2=Q/((2*(Q↑2)-G)*6.28*FO*C)
142 R2=R2*1000
143 R1=INT(R1*100+.5)/100:R2=INT(R2*100+.5)/100
144 R3=INT(R3*100+.5)/100
145 PRINT:PRINT"THEN:":PRINT" R1"," R2"," R3"
146 PRINT R1,R2,R3,"K-OHMS"
147 PRINT:INPUT"TRY ANOTHER SET OF VALUES (YES/NO) ";X$
148 IF X$="YES" THEN 129  ELSE 149
```

Continued on next page.

Fig. 6-22. Listing for BNDPASS1 program.

```
149 PRINT:PRINT"ENTER STANDARD COMPDNENT VALUES FDR:"
150 PRINT:INPUT"R1 (IN K-DHMS) ";R1
151 INPUT"R2 (IN K-OHMS) ";R2
152 INPUT"R3 (IN K-OHMS) ";R3
153 INPUT"C (IN UF) ";C
154 PRINT:PRINT:PRINT"BASED DN ABOVE STANDARD CDMPDNENTS CHOSEN:"
155 R3=1E3*R3:R2=1E3*R2:R1=1E3*R1:C=1E-6*C:Y=1/(R3*C↑2)
156 R1=1/R1:R2=1/R2:FD=SQR(Y*(R1+R2)):FO=FO/6.28
157 G=R3*R1/2
158 Q=G*6.28*FO*C/R1
159 PRINT"CENTER FREQUENCY =";INT(FO*10.+.5)/10;"HZ"
160 PRINT"CENTER FREQUENCY VDLTAGE GAIN =";INT(G*100+.5)/100
161 PRINT"Q =";INT(Q*100+.5)/100,"BANDWIDTH =";INT((FD/Q)*100+.5)/100;"HZ"
162 PRINT:INPUT"TRY ANOTHER SET OF STANDARD VALUES (YES/ND) ";X$
163 IF X$="YES" THEN 150  ELSE 164
164 CLS:INPUT"DO YOU WISH TD KNOW THE AMPLITUDE RESPONSE (YES/NO) ";X$
165 IF X$="YES" THEN 166  ELSE 172
166 INPUT"ENTER INPUT FREQUENCY IN HZ ";FI
167 F=FI/FO:FX=(((F↑2)-1)/F)↑2:V=1+(Q↑2)*FX
168 DB=(20*LOG(G)/LOG(10))-10*LOG(V)/LOG(10)
169 PRINT"   AMPLITUDE RESPONSE = "-:PRINTUSING G$;DB;:PRINT" DB"
170 PRINT:INPUT"TRY ANDTHER FREQUENCY (YES/ND) ";X$
171 IF X$="YES" THEN 166  ELSE 172
172 PRINT"DESIGN COMPLETED"
173 END
174 PRINT@Z9,CHR$(143);:L=1:  ' BLINKING CURSDR RDUTINE
175 Z9$=INKEY$
176 IFZ9$<>"" THEN PRINT @ Z9," ";:RETURN
177 L=L+1
178 IF L<10 THEN 175
179 PRINT@Z9," ";
180 FDR L=1 TO 5:NEXT
181 GOTO 174
```

Fig. 6-22 (cont). Listing for BNDPASS1 program.

```
          BANDPASS FILTER DESIGN FOR Q LESS THAN 10


DESIGN BASED DN KNOWING: 1. Q
                         2. 3-DB BANDWIDTH
                         3. UPPER & LOWER 3-DB FREQUENCIES

                                SELECT #  ? 1

Q ? 4.2

HIT <ENTER> TO CONTINUE ?
STANDARD CAPACITDR VALUE IN UF ? 0.01
CENTER FREQENCY IN HZ ? 750
CENTER FREQUENCY VOLTAGE GAIN ? 1.3

THEN:
 R1               R2               R3
 68.59            2.62             178.34          K-OHMS

TRY ANOTHER SET DF VALUES (YES/NO) ? ND
ENTER STANDARD COMPONENT VALUES FOR:
```

Fig. 6-23. Results for Example 6-4.

```
R1 (IN K-OHMS) ? 68.1
R2 (IN K-OHMS) ? 2.61
R3 (IN K-OHMS) ? 178
C (IN UF) ? 0.01

BASED ON ABOVE STANDARD COMPONENTS CHOSEN:
CENTER FREQUENCY = 752.8 HZ
CENTER FREQUENCY VOLTAGE GAIN = 1.31
Q = 4.21                    BANDWIDTH = 178.92 HZ


TRY ANOTHER SET OF STANDARD VALUES (YES/NO) ? NO
DO YOU WISH TO KNOW THE AMPLITUDE RESPONSE (YES/NO) ? YES
ENTER INPUT FREQUENCY IN HZ ? 400
   AMPLITUDE RESPONSE =  -12.90 DB

TRY ANOTHER FREQUENCY (YES/NO) ? YES
ENTER INPUT FREQUENCY IN HZ ? 1125
   AMPLITUDE RESPONSE =   -8.83 DB

TRY ANOTHER FREQUENCY (YES/NO) ? NO

DESIGN COMPLETED
READY
>
```

Fig. 6-23 (cont). Results for Example 6-4.

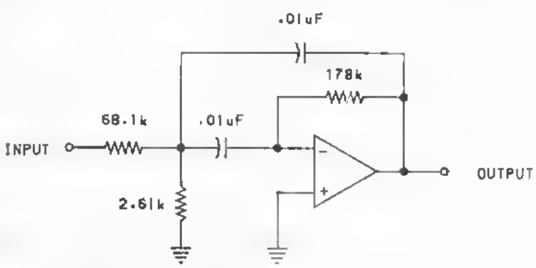


Fig. 6-24. Final circuit for Example 6-4.

A BANDPASS FILTER FOR Q LESS THAN 50

The simple multiple-feedback circuit of Fig. 6-21 is not stable for values of Q greater than 10. However, relatively simple bandpass filters having Q values between 10 and 50 are possible with the addition of a second op-amp and three resistors, as shown in Fig. 6-25. Using the BNDPASS2 program of Fig. 6-26, the design of the filter is simplified by making both capacitors equal, as before. In addition, because of the nature of the filter design, the center frequency voltage gain must be greater than:

$$G > (Q)^{1/2}$$

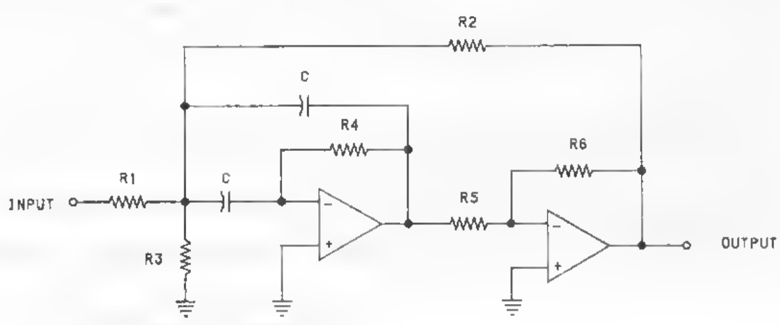
so that high gain accompanies high Q.

Example 6-5

Design a bandpass filter having a 3-dB bandwidth of 50 Hz at a center frequency of 1000 Hz (so that $Q = 20$).

Using a standard value of 0.018 μF and a center frequency voltage gain of 5, the output results are shown in Fig. 6-27. The final circuit is shown in Fig. 6-28.

Fig. 6-25. Bandpass filter for Q less than 50.



```
100 ' BANDPASS FILTER DESIGN FOR Q < 50 (BNDPASS2)
101 CLS:PRINTTAB(12)"BANDPASS FILTER DESIGN FOR Q LESS THAN 50"
102 PRINT
103 PRINT"DESIGN BASED ON KNOWING: 1. Q"
104 PRINT"                          2. 3-DB BANDWIDTH"
105 PRINT"                          3. UPPER & LOWER 3-DB FREQUENCIES"
106 INPUT"CHOICE";A
107 ON A GOTO 108  , 112  , 114
108 INPUT"Q ";Q
109 IF Q>=50 THEN 110  ELSE 111
110 PRINT"Q MUST BE LESS THAN 50.":GOTO 108
111 GOTO 120
112 INPUT"3-DB BANDWIDTH IN HZ ";BW
113 GOTO 120
114 INPUT"LOWER 3-DB FREQUENCY IN HZ ";FL
115 INPUT"UPPER 3-DB FREQUENCY IN HZ ";FH
116 FO=SQR(FH*FL):PRINT"   CENTER FREQUENCY =";INT(FO*10+.5)/10;"HZ"
117 BW=FH-FL:Q=FO/BW
118 PRINT"   3-DB BANDWIDTH =";INT(BW*10+.5)/10;"HZ"
119 PRINT"   Q=";INT(Q*100+.5)/100
120 PRINT:INPUT"HIT <ENTER> TO CONTINUE ";Z
121 CLS:INPUT"STANDARD CAPACITOR VALUE IN UF ";C
122 IF A=1 THEN 125  ELSE 123
123 IF A=3 THEN 128  ELSE 124
124 IF A=2 THEN 125
125 INPUT"CENTER FREQUENCY IN HZ ";FO
126 IF A=1 THEN 128  ELSE 127
127 Q=FO/BW
128 INPUT"CENTER FREQUENCY VOLTAGE GAIN ";G
129 IF G>SQR(Q) THEN 131
130 PRINT:PRINTTAB(10)"GAIN TOO LOW. TRY ANOTHER VALUE.":GOTO 128
131 K=G/SQR(Q):R1=Q*1000/(6.28*FO*C)
132 R2=R1*K*Q/((2*Q)-1)
133 R6=K*R1
134 DM=(Q↑2)-1-(2/K)+1/(K*Q):R3=R1/DM
135 R1=INT(R1*100+.5)/100:R2=INT(R2*100+.5)/100
136 R3=INT(R3*100+.5)/100:R4=INT(R4*100+.5)/100
137 R5=R1:R4=R1:R6=INT(R6*100+.5)/100
138 PRINT" R1"," R2"," R3"
139 PRINT R1,R2,R3,"K-OHMS":PRINT
140 PRINT" R4"," R5"," R6"
141 PRINT R4,R5,R6,"K-OHMS"
142 PRINT
143 INPUT"TRY ANOTHER SET OF VALUES (YES/NO) ";X$
144 IF X$="YES" THEN 121  ELSE 145
145 PRINT:PRINT"ENTER STANDARD COMPONENT VALUE FOR:"
146 INPUT"R1 (IN K-OHMS) ";R1
147 INPUT"R2 (IN K-OHMS) ";R2
148 INPUT"R3 (IN K-OHMS) ";R3
149 INPUT"R6 (IN K-OHMS) ";R6
150 INPUT"C (IN UF) ";C
151 PRINT:PRINT"BASED ON ABOVE STANDARD COMPONENTS CHOSEN:"
152 K=R6/R1:XX=(1000/(6.28*C*R1))*SQR((R1+2*R3)/R3)
153 FO=XX:Q=6.28*R1*FO*C/1000
154 G=K*SQR(Q)
155 PRINT"CENTER FREQUENCY = ";INT(FO*10+.5)/10;"HZ
156 PRINT"CENTER FREQUENCY GAIN = ";INT(G*100+.5)/100
157 PRINT"Q = ";INT(Q*100+.5)/100
158 PRINT:INPUT"TRY ANOTHER SET OF STANDARD VALUES (YES/NO) ";X$
159 IF X$="YES" THEN 145  ELSE 160
160 CLS:INPUT"DO YOU WISH TO KNOW THE AMPLITUDE RESPONSE (YES/NO) ";X$
```

Continued on next page.

Fig. 6-26. Listing for BNDPASS 2 program.

```
161 IF X$="YES" THEN 162  ELSE 168
162 INPUT"INPUT FREQUENCY IN HZ ";FI
163 F=FI/FO:FX=(((F↑2)-1)/F)↑2:V=1+(Q↑2)*FX
164 DB=(20*LOG(G)/LOG(10))-(10*LOG(V)/LOG(10))
165 PRINT"   AMPLITUDE RESPONSE = ";INT(DB*100+.5)/100;"DB"
166 PRINT:INPUT"TRY ANOTHER FREQUENCY (YES/NO) ";X$
167 IF X$="YES" THEN 162  ELSE 168
168 PRINT"DESIGN COMPLETED"
169 END
```

Fig. 6-26 (cont). Listing for BNDPASS2 program.

```
            BANDPASS FILTER DESIGN FOR Q LESS THAN 50

DESIGN BASED ON KNOWING: 1. Q
                         2. 3-DB BANDWIDTH
                         3. UPPER & LOWER 3-DB FREQUENCIES
CHOICE? 2
3-DB BANDWIDTH IN HZ ? 50

HIT <ENTER> TO CONTINUE ?
STANDARD CAPACITOR VALUE IN UF ? 0.018
CENTER FREQUENCY IN HZ ? 1000
CENTER FREQUENCY VOLTAGE GAIN ? 3

          GAIN TOO LOW. TRY ANOTHER VALUE.
CENTER FREQUENCY VOLTAGE GAIN ? 5
 R1             R2             R3
 176.93         101.44         .45            K-OHMS

 R4             R5             R6
 176.93         176.93         197.81         K-OHMS

TRY ANOTHER SET OF VALUES (YES/NO) ? NO


ENTER STANDARD COMPONENT VALUE FOR:
R1 (IN K-OHMS) ? 177
R2 (IN K-OHMS) ? 100
R3 (IN K-OHMS) ? 0.453
R6 (IN K-OHMS) ? 196
C (IN UF) ? 0.018

BASED ON ABOVE STANDARD COMPONENTS CHOSEN:
CENTER FREQUENCY =  990.5 HZ
CENTER FREQUENCY GAIN =  4.93
Q =  19.82

TRY ANOTHER SET OF STANDARD VALUES (YES/NO) ? NO
DO YOU WISH TO KNOW THE AMPLITUDE RESPONSE (YES/NO) ? YES
INPUT FREQUENCY IN HZ ? 990.5
   AMPLITUDE RESPONSE =  13.86 DB

TRY ANOTHER FREQUENCY (YES/NO) ? YES
INPUT FREQUENCY IN HZ ? 300
   AMPLITUDE RESPONSE = -21.62 DB

TRY ANOTHER FREQUENCY (YES/NO) ? NO

DESIGN COMPLETED
READY
>
```

Fig. 6-27. Results for Example 6-5.

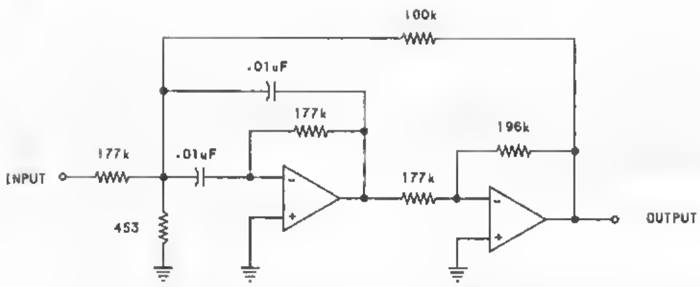


Fig. 6-28. Final circuit for Example 6-5.

```
100 ' STAGGERED-TUNED BUTTERWORTH BANDPASS FILTER DESIGN (TUNEDBP)
101 CLS:CLEAR:PRINT"STAGGERED TUNED BUTTERWORTH BANDPASS FILTER DESIGN"
102 PRINT:INPUT"NUMBER OF STAGES DESIRED ";P:IF P>4 GOTO 180
103 INPUT"LOWER 3-DB FREQUENCY (IN HZ) ";FL
104 INPUT"UPPER 3-DB FREQUENCY (IN HZ) ";FH
105 FZ=SQR(FL*FH):PRINT"   CENTER FREQUENCY =";INT(FZ*10+.5)/10;"HZ"
106 B=FH-FL:PRINT"   3-DB BANDWIDTH = ";INT(B*10+.5)/10;" HZ"
107 B1=B/FZ
108 IF P=2 THEN 109  ELSE 129
109 PRINT"1ST SECTION DESIGN PARAMETERS:"
110 A1=1+.365*B1:D1=.707*B1:F1=FZ*A1:F2=FZ/A1
111 Q1=1/D1
112 PRINT"   CENTER FREQUENCY = ";F1;" HZ     Q = ";INT(Q1*10+.5)/10
113 PRINT"2ND SECTION DESIGN PARAMETERS:"
114 PRINT"   CENTER FREQUENCY = ";F2;" HZ     Q = ";INT(Q1*10+.5)/10
115 A=-1:FI=FZ:GOTO 119
116 PRINT:INPUT"AMPLITUDE RESPONSE (1=YES, 0=NO) ";A
117 IF A=1 THEN 118  ELSE 128
118 INPUT"INPUT FREQUENCY IN HZ";FI
119 FA=FI/F1:FC=FI/F2:Q2=Q1
120 DA=SQR(1+(Q1↑2)*(((FA↑2)-1)/FA)↑2)
121 DC=SQR(1+(Q2↑2)*(((FC↑2)-1)/FC)↑2)
122 Y=1/(DA*DC):DB=8.68589*LOG(Y)
123 IF A=-1 THEN 124  ELSE 125
124 PRINT"GAIN OF CASCADED STAGES MUST BE AT LEAST";-INT(DB*10+.5)/10;"DB":GOTO 116
125 PRINT"    AMPLITUDE RESPONSE WITH INSERTION LOSS = ";INT(DB*10+.5)/10;"DB"
126 PRINT:INPUT"ANOTHER INPUT FREQUENCY (1=YES, 0=NO) ";C
127 IF C=1 THEN 118  ELSE 128
128 PRINT"DESIGN COMPLETED":GOTO 182
129 IF P=3 THEN 130  ELSE 152
130 PRINT"1ST SECTION DESIGN PARAMETERS:"
131 A1=1+.45*B1:A3=A1:D1=.5*B1:D3=D1:D2=1/B1:F2=FZ
132 F1=FZ*A1:F3=FZ/A3:Q1=1/D1:Q3=1/D3:Q2=D2
133 PRINT"   CENTER FREQUENCY = ";F1;"HZ     Q = ";INT(Q1*10+.5)/10
134 PRINT"2ND SECTION DESIGN PARAMETERS:"
135 PRINT"   CENTER FREQUENCY = ";F2;"HZ     Q = ";INT(Q2*10+.5)/10
136 PRINT"3RD SECTION DESIGN PARAMETERS:"
137 PRINT"   CENTER FREQUENCY = ";F3;"HZ     Q = ";INT(Q3*10+.5)/10
138 A=-1:FI=F2:GOTO 142
139 PRINT:INPUT"AMPLITUDE RESPONSE (1=YES, 0=NO) ";A
140 IF A=1 THEN 141  ELSE 151
141 INPUT"INPUT FREQUENCY IN HZ ";FI
142 FA=FI/F1:FB=FI/F2:FC=FI/F3:DA=SQR(1+(Q1↑2)*(((FA↑2)-1)/FA)↑2)
143 DC=SQR(1+(Q2↑2)*(((FB↑2)-1)/FB)↑2)
144 DD=SQR(1+(Q3↑2)*(((FC↑2)-1)/FC)↑2
145 Y=1/(DA*DC*DD):DB=8.68589*LOG(Y)
146 IF A=-1 THEN 147  ELSE 148
147 PRINT"GAIN OF CASCADED STAGES MUST BE AT LEAST";-INT(DB*10+.5)/10;"DB":GOTO 139
148 PRINT"   AMPLITUDE RESPONSE WITH INSERTION LOSS = ";INT(DB*10+.5)/10;"DB"
```

Continued on next page.

Fig. 6-29. Listing for TUNEDBP program.

```
149 PRINT:INPUT"ANOTHER INPUT FREQUENCY (1=YES, 0=NO) ";C
150 IF C=1 THEN 141  ELSE 151
151 PRINT"DESIGN COMPLETED":GOTO 182
152 IF P=4 THEN 153  ELSE 180
153 PRINT"1ST SECTION DESIGN PARAMETERS:"
154 A1=1+.485*B1:A3=1+.195*B1:D1=.38*B1:D3=.923*B1
155 F1=FZ*A1:F2=FZ*A3:F3=FZ/A3:F4=FZ/A1
156 Q1=1/D1:Q2=1/D3:Q3=Q2:Q4=Q1
157 PRINT"   CENTER FREQUENCY = ";F1;" HZ    Q = ";INT(Q1*10+.5)/10
158 PRINT"2ND SECTION DESIGN PARAMETERS:"
159 PRINT"   CENTER FREQUENCY = ";F2;" HZ    Q = ";INT(Q2*10+.5)/10
160 PRINT"3RD SECTION DESIGN PARAMETERS:"
161 PRINT"   CENTER FREQUENCY = ";F3;" HZ    Q = ";INT(Q3*10+.5)/10
162 PRINT"4TH SECTION DESIGN PARAMETERS:"
163 PRINT"   CENTER FREQUENCY = ";F4;" HZ    Q = ";INT(Q4*10+.5)/10
164 A=-1:FI=FZ:GOTO 168
165 PRINT:INPUT"AMPLITUDE RESPONSE (1=YES, 0=NO) ";A
166 IF A=0 THEN 179  ELSE 167
167 INPUT"INPUT FREQUENCY IN HZ ";FI
168 FA=FI/F1:FB=FI/F2:FC=FI/F3:FD=FI/F4
169 DA=SQR(1+(Q1↑2)*(((FA↑2)-1)/FA)↑2)
170 DC=SQR(1+(Q2↑2)*(((FB↑2)-1)/FB)↑2)
171 DD=SQR(1+(Q3↑2)*(((FC↑2)-1)/FC)↑2)
172 DE=SQR(1+(Q4↑2)*(((FD↑2)-1)/FD)↑2)
173 Y=1/(DA*DC*DD*DE):DB=8.68589*LOG(Y)
174 IF A=-1 THEN 175  ELSE 176
175 PRINT"GAIN OF CASCADED STAGES MUST BE AT LEAST";-INT(DB*10+.5)/10;"DB":GOTO 165
176 PRINT"   AMPLITUDE RESPONSE WITH INSERTION LOSS = ";INT(DB*10+.5)/10;"DB"
177 PRINT:INPUT"ANOTHER INPUT FREQUENCY (1=YES, 0=NO) ";C
178 IF C=1 THEN 167  ELSE 179
179 PRINT"DESIGN COMPLETED":GOTO 182
180 PRINT"DESIGNS LIMITED UP TO 4 STAGES - TRY AGAIN.
181 GOTO 102
182 END
```

Fig. 6-29 (cont). Listing for TUNEDBP program.

STAGGERED-TUNED BUTTERWORTH BANDPASS FILTERS

Very often it is required to have a bandpass filter that passes a range of frequencies, instead of a single frequency. In most cases a single stage bandpass section like those previously described will not offer the necessary response. Consequently, it may be necessary to cascade two or more bandpass sections in order to do the job. The program listing (TUNEDBP) shown in Fig. 6-29 allows for the design of either a 2, 3, or 4 stage Butterworth (maximally flat passband) bandpass filter. The program calculates the required center frequency and Q for each stage. In addition, the insertion loss of the filter is also determined, which must be made up when designing the individual stages. The following example illustrates the use of the TUNEDBP program.

Example 6-6

Design a 3-stage bandpass filter having a lower 3-dB frequency of 2050 Hz and an upper 3-dB frequency of 2350 Hz. When the TUNEDBP program is run, the results, as shown in Fig. 6-30, are obtained.

For our example, we see that the first and third sections have values of Q that are greater than 10 (i.e., 14.6). Consequently, we should use the filter circuit shown in Fig. 6-25 and the BNDPASS2 program to calculate each section separately. For the second section, the Q is less than 10 (i.e., 7.3), so that we are able to use the circuit of Fig. 6-21 and the corresponding BNDPASS1 program. If the Q of any stage is found to be greater than 50, then the state-variable filter should be used. In addition, the overall gain of the three stages must be at least 12.16 dB, or a voltage gain of 4.05 in order to overcome the insertion loss presented by the filter.

Fig. 6-31 shows one of many circuits that are possible to satisfy our design. The voltage gain for the first and third sections is 4, while the gain for the second section is 2, giving an overall gain of 32 (+30.1 dB).

```
STAGGERED TUNED BUTTERWORTH BANDPASS FILTER DESIGN

NUMBER OF STAGES DESIRED ? 3
LOWER 3-DB FREQUENCY (IN HZ) ? 2050
UPPER 3-DB FREQUENCY (IN HZ) ? 2350
   CENTER FREQUENCY = 2194.9 HZ
   3-DB BANDWIDTH =  300  HZ
1ST SECTION DESIGN PARAMETERS:
   CENTER FREQUENCY =  2329.88 HZ     Q =  14.6
2ND SECTION DESIGN PARAMETERS:
   CENTER FREQUENCY =  2194.88 HZ     Q =  7.3
3RD SECTION DESIGN PARAMETERS:
   CENTER FREQUENCY =  2067.7 HZ     Q =  14.6
GAIN OF CASCADED STAGES MUST BE AT LEAST 12.2 DB


AMPLITUDE RESPONSE (1=YES, 0=NO) ? 1
INPUT FREQUENCY IN HZ ? 1750
   AMPLITUDE RESPONSE WITH INSERTION LOSS = -43.5 DB

ANOTHER INPUT FREQUENCY (1=YES, 0=NO) ? 1
INPUT FREQUENCY IN HZ ? 3000
   AMPLITUDE RESPONSE WITH INSERTION LOSS = -52.1 DB

ANOTHER INPUT FREQUENCY (1=YES, 0=NO) ? 0

DESIGN COMPLETED

READY
>_
```

Fig. 6-30. Results for Example 6-6.

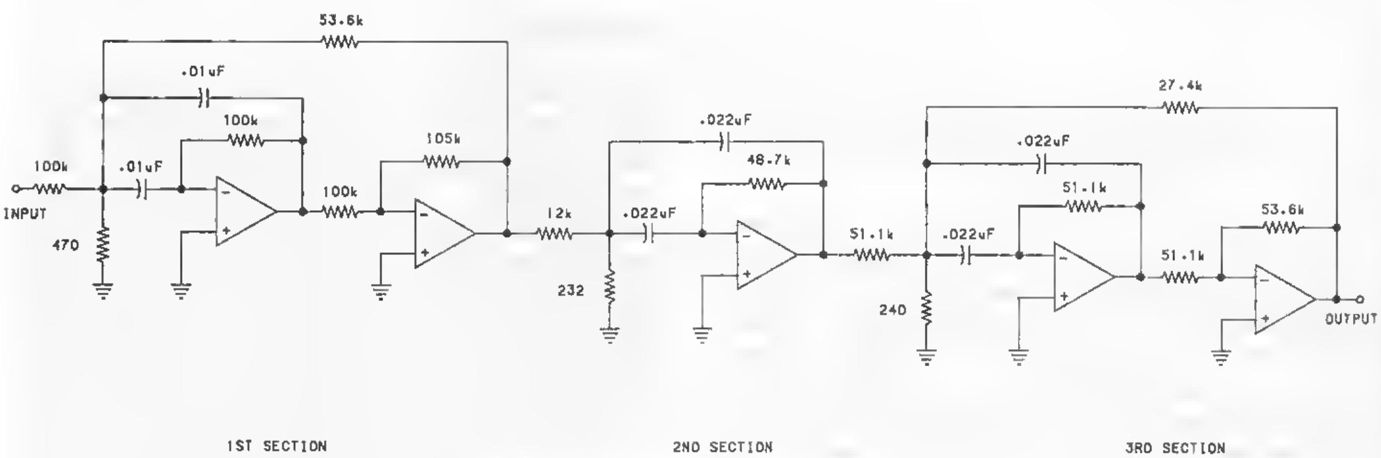


Fig. 6-31. Final circuit for Example 6-6.

THE NOTCH FILTER

Fig. 6-32 shows a notch filter for values of Q less than 25. For values of Q greater than 25, the state-variable notch filter should be used. As with all the filter designs discussed, both capacitors are made equal to each other.

The NOTCH program listed in Fig. 6-33 requires the following initial information:

1. Design determined by knowing either Q, 3-dB bandwidth, or the 3-dB frequencies
2. Standard capacitor value
3. Notch, or null frequency
4. Passband voltage gain

after which the remaining resistor values are determined.

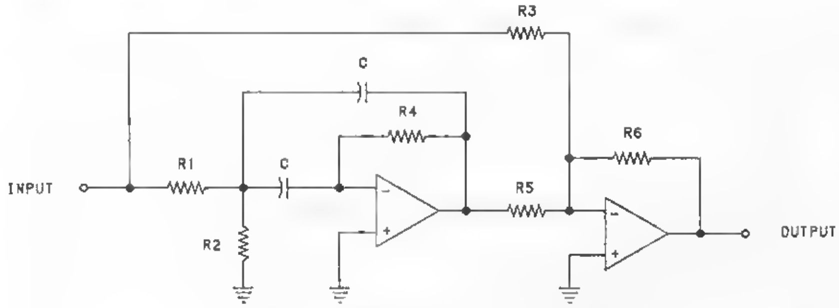


Fig. 6-32. Notch filter.

```
100 ' NOTCH FILTER DESIGN (NOTCH)
101 CLS:PRINT,"NOTCH FILTER DESIGN"
102 PRINT
103 PRINT"DESIGN BASED ON KNOWING: 1. Q"
104 PRINT"                         2. 3-DB BANDWIDTH"
105 PRINT"                         3. UPPER & LOWER 3-DB FREQUENCIES"
106 PRINT@473,"SELECT # ":Z9=483:GOSUB171
107 IF Z9$<>"1" AND Z9$<>"2" AND Z9$<>"3" THEN 106  ELSE 108
108 PRINT@483,Z9$:IF Z9$="1" THEN 111
109 IF Z9$="2" THEN 115
110 IF Z9$="3" THEN 117
111 PRINT:INPUT"Q ";Q
112 IF Q>=25 THEN 113  ELSE 114
113 PRINT"Q MUST BE LESS THAN 25.":GOTO 111
114 GOTO 124
115 PRINT:INPUT"3-DB BANDWIDTH IN HZ ";BW
116 GOTO 124
117 PRINT:INPUT"LOWER 3-DB FREQUENCY IN HZ ";FL
118 INPUT"UPPER 3-DB FREQUENCY IN HZ ";FH
119 FO=SQR(FH*FL)
120 PRINT"   NOTCH (NULL) FREQUENCY =";INT(FO*10+.5)/10;"HZ"
121 BW=FH-FL:Q=FO/BW
122 PRINT"   3-DB BANDWIDTH =";INT(BW*10+.5)/10;"HZ"
123 PRINT"   Q =";INT(Q*10+.5)/10
124 PRINT:INPUT"HIT <ENTER> TO CONTINUE ";Z
125 CLS:INPUT"STANDARD CAPACITOR VALUE IN UF ";C
126 IF Z9$="1" THEN 129  ELSE 127
127 IF Z9$="3" THEN 132  ELSE 128
128 IF Z9$="2" THEN 129
129 INPUT"NOTCH (NULL) FREQENCY IN HZ ";FO
130 IF Z9$="1" THEN 132  ELSE 131
131 Q=FO/BW
132 INPUT"PASSBAND VOLTAGE GAIN ";G
133 H=100/(FO*C):R1=0.796*Q*H:R2=R1/((Q+2)-1)
134 R3=H:R4=4*R1:R5=2*H:R6=G*H
135 R1=INT(R1*100+.5)/100:R2=INT(R2*100+.5)/100
136 R3=INT(R3*100+.5)/100
137 R4=INT(R4*100+.5)/100:R5=INT(R5*100+.5)/100
138 R6=INT(R6*100+.5)/100
139 PRINT:PRINT"THEN:":PRINT" R1"," R2"," R3"
140 PRINT R1,R2,R3,"K-OHMS"
141 PRINT:PRINT" R4"," R5"," R6"
142 PRINT R4,R5,R6,"K-OHMS"
143 PRINT:INPUT"TRY ANOTHER SET OF VALUES (YES/NO) ";X$
144 IF X$="YES" THEN 125  ELSE 145
```

Continued on next page.

Fig. 6-33. Listing for NOTCH program.

```
145 PRINT:PRINT"ENTER STANDARD COMPONENT VALUES FOR:"
146 PRINT:INPUT"R1 (IN K-OHMS) ";R1
147 INPUT"R2 (IN K-OHMS) ";R2
148 INPUT"R3 (IN K-OHMS) ";R3
149 INPUT"R4 (IN K-OHMS) ";R4
150 INPUT"R6 (IN K-OHMS) ";R6
151 INPUT"C (IN UF) ";C
152 PRINT:PRINT"BASED ON ABOVE STANDARD COMPONENTS CHOSEN:"
153 R2=1E3*R2:R1=1E3*R1:C=1E-6*C:R4=1E3*R4
154 FO=SQR(((R2+R1)/(R1*R2))*(1/(R4*(C↑2))))/6.28
155 H=1E-4/(FO*C):Q=R1*1E-3/(.796*H):G=R6/R3
156 PRINT"NOTCH FREQUENCY =";INT(FO*100.+.5)/100;"HZ"
157 PRINT"PASSBAND VOLTAGE GAIN =";INT(G*100+.5)/100
158 PRINT"Q =";INT(Q*100+.5)/100,"BANDWIDTH =";INT((FO/Q)*100+.5)/100;"HZ"
159 PRINT:INPUT"TRY ANOTHER SET OF STANDARD VALUES (YES/NO) ";X$
160 IF X$="YES" THEN 146  ELSE 161
161 CLS:INPUT"DO YOU WISH TO KNOW THE AMPLITUDE RESPONSE (YES/NO) ";X$
162 IF X$="YES" THEN 163  ELSE 169
163 INPUT"ENTER INPUT FREQUENCY IN HZ ";FI
164 F=FI/FO:V=4.34294*LOG((1-F↑2)↑2)-4.34294*LOG((1-F↑2)↑2+F/Q↑2)
165 DB=(20*LOG(G)/LOG(10))+V
166 PRINT"   AMPLITUDE RESPONSE = ";DB;:PRINT" DB"
167 PRINT:INPUT"TRY ANOTHER FREQUENCY (YES/NO) ";X$
168 IF X$="YES" THEN 163  ELSE 169
169 PRINT"DESIGN COMPLETED"
170 END
171 PRINT@Z9,CHR$(143);:L=1:    'BLINKING CURSOR ROUTINE
172 Z9$=INKEY$
173 IFZ9$<>"" THEN PRINT @ Z9," ";:RETURN
174 L=L+1
175 IF L<10 THEN 172
176 PRINT@Z9," ";
177 FOR L=1 TO 5:NEXT
178 GOTO 171
```

Fig. 6-33 (cont). Listing for NOTCH program.

Example 6-7

Design a 60-Hz notch filter with a Q of 20 and a passband gain of 4 (+12.04 dB).

Selecting a standard capacitor value of 0.22 μF, Fig. 6-34 shows the program output, while Fig. 6-35 illustrates the final circuit.

A WIDEBAND FILTER

Very often, particularly in audio applications, it is desired to pass a wide band of frequencies such that the bandwidth of the filter is to be more than twice its center frequency. In addition, the passband gain is to be relatively constant. Such is the case of a filter that passes audio frequencies ranging from 300 Hz to 3000 Hz.

The "center frequency" is then 949 Hz, while the bandwidth is 2700 Hz. Virtually no standard bandpass filter (including staggered-tuned designs) will function properly over this range. However, by cascading a 3000-Hz low-pass section with a 300-Hz high-pass section (from second to sixth order), we are then able to produce the desired response, as shown in the following example.

Example 6-8

Design a 300- to 3000-Hz wideband filter having relatively constant passband gain with second-order stopband responses.

Since the passband gain is to be essentially constant, then a second-order Butterworth design should be used for each section. Using the LPHP program discussed earlier in this chapter, we can then design each section individually, as shown by the final circuit of Fig. 6-36.

Fig. 6-34. Results for Example 6-7.

```
                  NOTCH FILTER DESIGN
DESIGN BASED ON KNOWING: 1. Q
                         2. 3-DB BANDWIDTH
                         3. UPPER & LOWER 3-DB FREQUENCIES
SELECT # ? 1

Q ? 20

HIT <ENTER> TO CONTINUE ?

STANDARD CAPACITOR VALUE IN UF ? 0.22
NOTCH (NULL) FREQENCY IN HZ ? 60
PASSBAND VOLTAGE GAIN ? 4

THEN:
 R1              R2              R3
 120.61          .3              7.58            K-OHMS

 R4              R5              R6
 482.42          15.15           30.3            K-OHMS

TRY ANOTHER SET OF VALUES (YES/NO) ? NO

ENTER STANDARD COMPONENT VALUES FOR:

R1 (IN K-OHMS) ? 120
R2 (IN K-OHMS) ? .300
R3 (IN K-OHMS) ? 7.5
R4 (IN K-OHMS) ? 487
R6 (IN K-OHMS) ? 30
C (IN UF) ? 0.22

BASED ON ABOVE STANDARD COMPONENTS CHOSEN:
NOTCH FREQUENCY = 59.96 HZ
PASSBAND VOLTAGE GAIN = 4
Q = 19.89                 BANDWIDTH = 3.02 HZ

TRY ANOTHER SET OF STANDARD VALUES (YES/NO) ? NO
DO YOU WISH TO KNOW THE AMPLITUDE RESPONSE (YES/NO) ? YES
ENTER INPUT FREQUENCY IN HZ ? 50
   AMPLITUDE RESPONSE =  11.9436  DB

TRY ANOTHER FREQUENCY (YES/NO) ? YES
ENTER INPUT FREQUENCY IN HZ ? 59.96
   AMPLITUDE RESPONSE = -40.2648  DB

TRY ANOTHER FREQUENCY (YES/NO) ? NO
DESIGN COMPLETED
READY
>
```

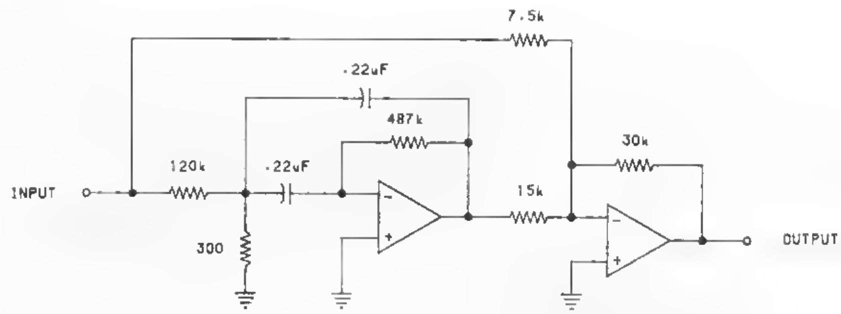


Fig. 6-35. Final circuit for Example 6-7.

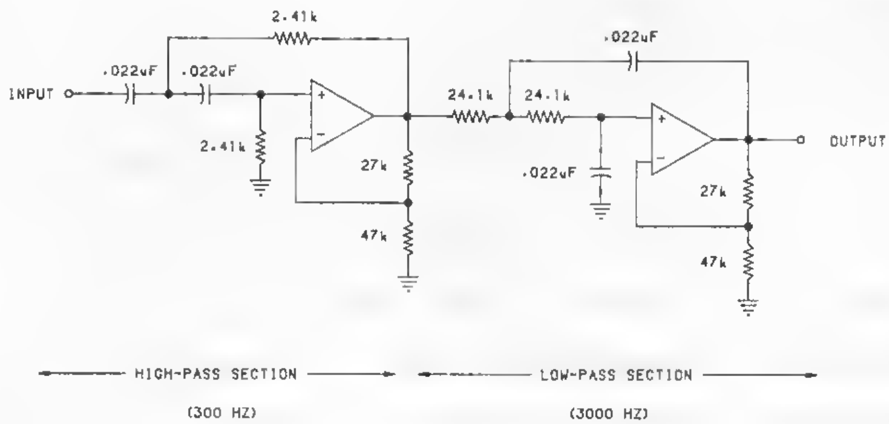


Fig. 6-36. Final circuit for Example 6-8.

7

Solid-State Devices

This chapter describes programs which simplify the design of zener diode voltage regulators, the single stage common-emitter transistor amplifier, the analysis of transistor biasing, heat sinks, and 555 timer circuits.

ZENER DIODE VOLTAGE REGULATOR DESIGN

The ZENER program listing of Fig. 7-1 allows for the determination of the proper resistance and zener diode power rating for the regulator circuit of Fig. 7-2. When executed, the program requires the following specifications:

1. Minimum and maximum dc input voltage
2. Minimum and maximum load current, in milliamperes
3. Zener diode (regulated output) voltage

In addition to determining the series resistance and the minimum power rating for the zener diode, the program also determines the diode rating if the load is disconnected from the regulator, so that the load current is zero.

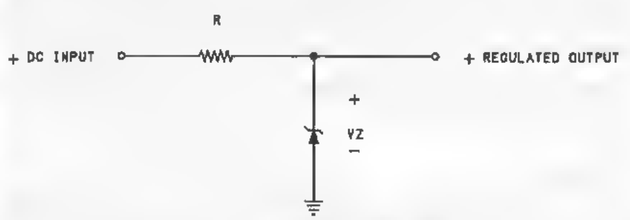


Fig. 7-2. Zener diode voltage regulator.

Example 7-1

Determine the necessary series resistance and diode rating for a regulator circuit having a dc input voltage from 11.95 to 12.05 volts, and connected to a load which draws between 75 and 300 mA.

The output results are shown in Fig. 7-3.

```
100 'ZENER DIODE VOLTAGE REGULATOR DESIGN (ZENER)
101 CLS:PRINTTAB(10)"ZENER DIODE VOLTAGE REGULATOR DESIGN"
102 PRINT:INPUT"MINIMUM DC INPUT VOLTAGE ";VN
103 INPUT"MAXIMUM DC INPUT VOLTAGE ";VX
104 INPUT"MINIMUM LOAD CURRENT (IN MA) ";IN
105 INPUT"MAXIMUM LOAD CURRENT (IN MA) ";IX
106 IN=IN*1E-3:IX=IX*1E-3
107 INPUT"ZENER DIODE VOLTAGE ";VZ
108 R=(VN-VZ)/(1.1*IX)
109 P=VZ*(((VX-VZ)/R)-IN)
110 R=INT(R*10+.5)/10:P=INT(P*100+.5)/100
111 PRINT:PRINT"THEN:"
112 PRINT:PRINT"REQUIRED SERIES RESISTANCE =";R;"OHMS"
113 PRINT"MINIMUM ZENER DIODE RATING =";P;"WATTS"
114 IF IN=0 THEN 117
115 Z=VZ*((VX-VZ)/R):Z=INT(Z*100+.5)/100
116 PRINT"MINIMUM ZENER DIODE RATING WITH NO LOAD =";Z;"WATTS"
117 PRINT:END
```

Fig. 7-1. Listing for ZENER program.

```
              ZENER DIODE VOLTAGE REGULATOR DESIGN

MINIMUM DC INPUT VOLTAGE ? 11.95
MAXIMUM DC INPUT VOLTAGE ? 12.05
MINIMUM LOAD CURRENT (IN MA) ? 75
MAXIMUM LOAD CURRENT (IN MA) ? 300
ZENER DIODE VOLTAGE ? 5.1

THEN:

REQUIRED SERIES RESISTANCE = 20.8 OHMS
MINIMUM ZENER DIODE RATING = 1.33 WATTS
MINIMUM ZENER DIODE RATING WITH NO LOAD = 1.7 WATTS


READY
>_
```

Fig. 7-3. Results for Example 7-1.

555 TIMER DESIGN

The 555 integrated circuit timer is one of the most popular devices in use. With the same device, it is possible to design both monostable (one-shot) and astable (free-running) multivibrator circuits, as shown in Figs. 7-4 and 7-5, respectively.

The TIMER555 program listing of Fig. 7-6 determines the required component values for monostable and astable circuits, as well as displaying the final schematic circuit on the video display. Depending on the type of design, the following initial information is required:

1. Monostable Design:
    a. Standard capacitor value
    b. Time delay in seconds
2. Astable Design:
    a. Standard capacitor value
    b. Output frequency in Hz
    c. Percent duty cycle

The percent duty cycle must be greater than 50% and less than 100%. If the duty cycle is to be symmetrical, then use the value of 50.1% for the duty cycle. For both designs, the program permits the calculated component values to be changed if they are not close enough to standard values.

Example 7-2

Using a 555 timer, design a monostable multivibrator having a time delay of 0.055 second.

As shown in Fig. 7-7A, we started by choosing a value of 0.01 μF. However the computed resistance value did not appear to be close to a standard value. The process was repeated until the final value of 0.022 μF was used so that a 2.2-megohm resistor could be used. The final circuit is shown in Fig. 7-7B.

Example 7-3

Using a 555 timer, design an astable multivibrator with an output frequency of 2400 Hz and a 55% duty cycle. Repeat for a 95% duty cycle.

For a 55% duty cycle, the output results are shown in Fig. 7-8, while Fig. 7-9 shows the results for a 95% duty cycle.

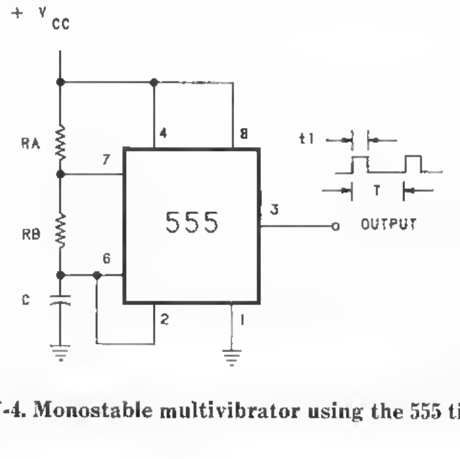


Fig. 7-4. Monostable multivibrator using the 555 timer.

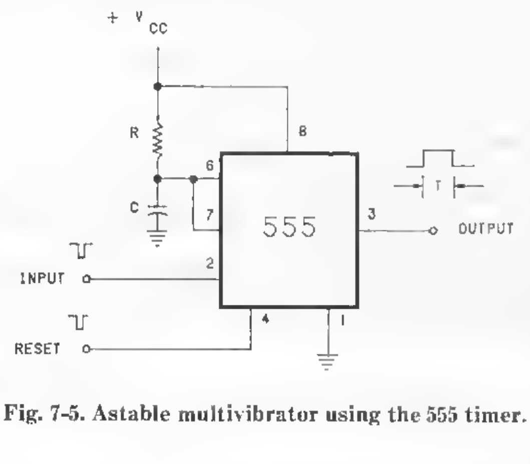


Fig. 7-5. Astable multivibrator using the 555 timer.

```
100 ' 555 TIMER DESIGN - WITH GRAPHICS (TIMER555)
101 CLS:PRINTTAB(17)"555 INTEGRATED CIRCUIT TIMER DESIGN"
102 PRINT:INPUT"<M>ONOSTABLE OR <A>STABLE MODE ";C$
103 IF C$<>"M" AND C$<>"A" THEN 101
104 IF C$="M" THEN 105  ELSE 114
105 CLS:PRINTTAB(16)"555 TIMER - MONOSTABLE OPERATION"
106 PRINT:INPUT"TIME DELAY (SECONDS) ";T
107 IF T<1E-3 THEN 106
108 INPUT"C (UF) ";C
109 IF C<.001 THEN 108
110 R=INT(((1E3*T)/(1.1*C))*100+.5)/100:PRINT"     R =";R;"K-OHMS"
111 PRINT:INPUT"CHANGE VALUES (YES/NO) ";A$:IF A$="YES" THEN 108
112 GOSUB 131
113 END
114 CLS:PRINTTAB(18)"555 TIMER - ASTABLE OPERATION"
115 PRINT:INPUT"OUTPUT FREQUENCY (HZ) ";F
116 IF F>200E3 THEN 115
117 INPUT"PERCENT DUTY CYCLE (>.50%) ";D:D=D/100
118 IF D<=.5 THEN 117
119 INPUT"C (UF) ";C:C=C*1E-6
120 IF C<1E-9 THEN 119
121 RA=(2*D-1)/(LOG(2)*C*F)
122 RB=(1/(2*LOG(2)*C*F))-(RA/2)
123 X=C*(RA+2*RB):F=(1/LOG(2))*1/X
124 RA=INT((RA/1E3)*100+.5)/100:RB=INT((RB/1E3)*100+.5)/100:C=C/1E-6
125 D=D*100:PRINT"     RA =";RA;"K-OHMS";"     RB =";RB;"K-OHMS"
126 PRINT:INPUT"CHANGE INPUT VALUES (YES/NO) ";B$
127 IF B$="NO" THEN 128  ELSE 117
128 GOSUB 170
129 GOSUB 182
130 END
131 CLS:PRINT@0,"FINAL DESIGN":PRINT@20,"+VCC"
132 PRINT@35,"+VCC"
133 PRINT@159,"4      8"
134 PRINT@210,"R":PRINT@262,R;"K-OHMS"
135 PRINT@282,"6"
136 PRINT@362,"3"
137 PRINT@417,"555"
138 PRINT@433,"OUTPUT":PRINT@496,T;"SEC"
139 PRINT@529,"C +":PRINT@538,"7":PRINT@585,C;"UF":PRINT@671,"2       1"
140 PRINT@848,"INPUT"
141 FORX=58TO81:SET(X,11):SET(X,26):NEXT
142 FORX=57TO82:SET(X,10):SET(X,27):NEXT
143 FORY=11TO26:SET(58,Y):SET(81,Y):NEXT
144 FORY=10TO27:SET(57,Y):SET(82,Y):NEXT
145 FORY=27TO33:SET(74,Y):NEXT
146 FORX=71TO77:SET(X,33):NEXT
147 SET(71,34):SET(74,34):SET(77,34)
148 FORY=27TO39:SET(65,Y):NEXT
149 FORX=45TO65:SET(X,39):NEXT
150 FORX=83TO95:SET(X,19):NEXT
151 FORY=4TO10:SET(74,Y):NEXT
152 FORY=7TO10:SET(65,Y):NEXT
153 FORX=65TO74:SET(X,7):NEXT
154 FORX=45TO57:SET(X,17):SET(X,21):NEXT
155 FORY=14TO25:SET(44,Y):NEXT
156 FORX=41TO47:SET(X,26):SET(X,28):NEXT
157 FORY=29TO33:SET(44,Y):NEXT
158 FORX=41TO47:SET(X,34):NEXT
159 SET(41,35):SET(44,35):SET(47,35)
160 FORX=43TO45:SET(X,13):SET(X,10):NEXT
```

Continued on next page.

Fig. 7-6. Listing for TIMER555 program.

```
161 FORY=10TO13:SET(43,Y):SET(45,Y):NEXT
162 FORY=4TO10:SET(44,Y):NEXT
163 SET(25,38):SET(26,38):SET(26,39):SET(26,40):SET(27,40)
164 SET(28,40):SET(28,39):SET(28,38):SET(29,38)
165 SET(98,16):SET(99,16):SET(100,16):SET(100,15):SET(100,14)
166 SET(105,16):SET(106,16):SET(107,16)
167 FORX=100TO105:SET(X,14):NEXT
168 SET(105,15)
169 RETURN
170 CLS:PRINT@0,"FINAL DESIGN":PRINT@18,"+VCC"
171 PRINT@35,"+VCC"
172 PRINT@142,"RA":PRINT@159,"4      8"
173 PRINT@197,RA;"K-OHMS":PRINT@334,"RB"
174 PRINT@346,"7":PRINT@362,"3"
175 PRINT@389,RB;"K-OHMS":PRINT@417,"555"
176 PRINT@433,"OUTPUT"
177 PRINT@474,"6":PRINT@496,F;"HZ"
178 PRINT@556,"DUTY CYCLE =";D;"%":PRINT@590,"C  +"
179 PRINT@646,C;"UF"
180 PRINT@671,"2      1"
181 PRINT@832,"":RETURN
182 FORX=58TO81:SET(X,11):SET(X,26):NEXT
183 FORY=11TO26:SET(58,Y):SET(81,Y):NEXT
184 FORX=57TO82:SET(X,10):SET(X,27):NEXT
185 FORY=10TO27:SET(57,Y):SET(82,Y):NEXT
186 FORY=27TO37:SET(74,Y):NEXT
187 FORX=71TO77:SET(X,37):NEXT
188 SET(71,38):SET(74,38):SET(77,38)
189 FORY=27TO34:SET(65,Y):NEXT
190 FORX=83TO95:SET(X,19):NEXT
191 FORY=4TO10:SET(74,Y):NEXT
192 FORX=65TO74:SET(X,7):NEXT
193 FORY=7TO10:SET(65,Y):NEXT
194 FOR X=40TO57:SET(X,13):SET(X,24):NEXT
195 FORX=52TO64:SET(X,34):NEXT
196 FORY=25TO33:SET(52,Y):NEXT
197 FORY=4TO6:SET(40,Y):NEXT
198 FORY=10TO16:SET(40,Y):NEXT
199 FORY=20TO28:SET(40,Y):NEXT
200 FORX=39TO41:SET(X,6):SET(X,10):SET(X,16):SET(X,20):NEXT
201 FORY=6TO10:SET(39,Y):SET(41,Y):NEXT
202 FORY=16TO20:SET(39,Y):SET(41,Y):NEXT
203 FORY=30TO38:SET(40,Y):NEXT
204 FORX=37TO43:SET(X,37):NEXT
205 SET(37,38):SET(43,38)
206 FORX=37TO43:SET(X,28):SET(X,30):NEXT
207 RETURN
```

Fig. 7-6 (cont). Listing for TIMER555 program.

TRANSISTORS

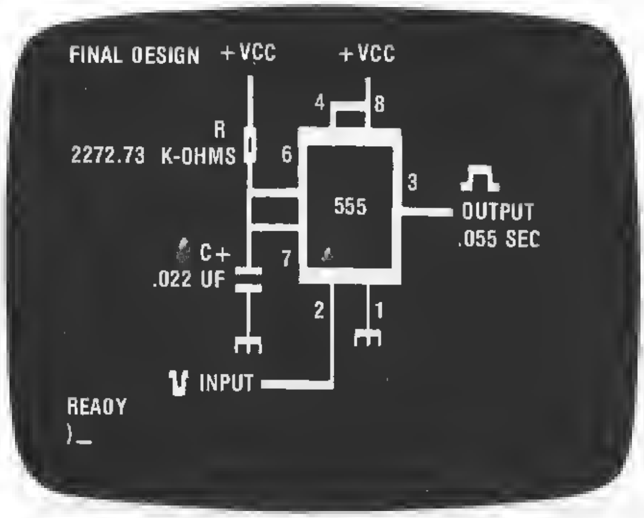
In these days of integrated circuits that virtually do everything, the transistor is still an important facet of the design and analysis of electronic circuits. In general, the one-power supply bias network of a bipolar transistor will have the generalized circuit of Fig. 7-10, in that one or more resistors will be either open or short circuited.

For any given circuit, the BIAS program of Fig. 7-11 will determine the operating point of the transistor with the values of the three terminal currents, and the quiescent output voltage, V_{CO}. When run, the required parameters are:

1. Supply voltage
2. Typical beta of the transistor
3. Leakage current I_{ceo}
4. The network resistors (in ohms)

```
555 TIMER - MONOSTABLE OPERATION


TIME DELAY (SECONDS) ? 0.055
C (UF) ? .01
     R = 5000 K-OHMS


CHANGE VALUES (YES/NO) ? YES
C (UF) ? .018
     R = 2777.78 K-OHMS


CHANGE VALUES (YES/NO) ? YES
C (UF) ? .022
     R = 2272.73 K-OHMS


CHANGE VALUES (YES/NO) ? NO
```

(A) Output results.

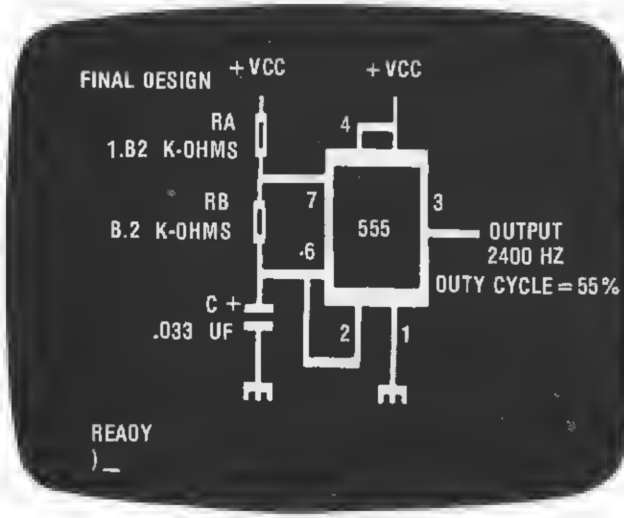


(B) Video plot.

Fig. 7-7. Results for Example 7-2.

```
555 TIMER - ASTABLE OPERATION

OUTPUT FREQUENCY (HZ) ? 2400
PERCENT DUTY CYCLE (> 50%) ? 55
C (UF) ? .001
     RA = 60.11 K-OHMS RB = 270.51 K-OHMS

CHANGE INPUT VALUES (YES/NO) ? YES
PERCENT DUTY CYCLE (> 50%) ? 55
C (UF) ? .033
     RA = 1.82 K-OHMS    RB = 8.2 K-OHMS

CHANGE INPUT VALUES (YES/NO) ? NO
```

(A) Output results.

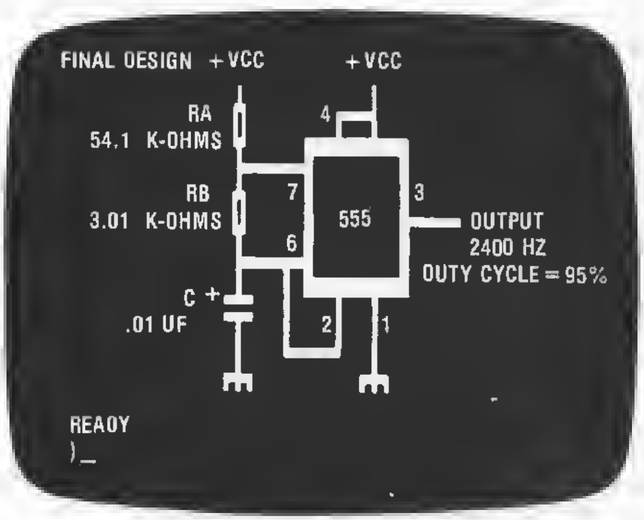


(B) Video plot.

Fig. 7-8. Results for Example 7-3 for a 55% duty cycle.

```
555 TIMER - ASTABLE OPERATION

OUTPUT FREQUENCY (HZ) ? 2400
PERCENT DUTY CYCLE (> 50%) ? 95
C (UF) ? .033
     RA = 16.39 K-OHMS   RB = .91 K-OHMS

CHANGE INPUT VALUES (YES/NO) ? YES
PERCENT DUTY CYCLE (> 50%) ? 95
C (UF) ? .018
     RA = 30.06 K-OHMS   RB = 1.67 K-OHMS

CHANGE INPUT VALUES (YES/NO) ? NO
```

(A) Output results.



(B) Video plot.

Fig. 7-9. Results for Example 7-3 for a 95% duty cycle.

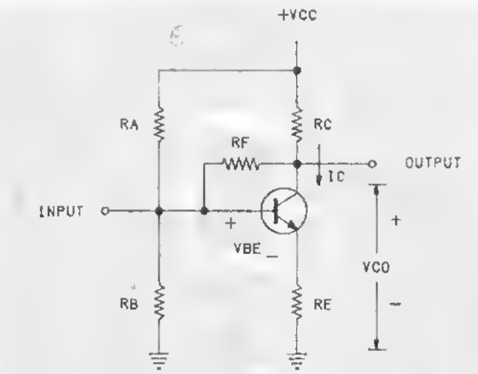


Fig. 7-10. Generalized transistor bias circuit.

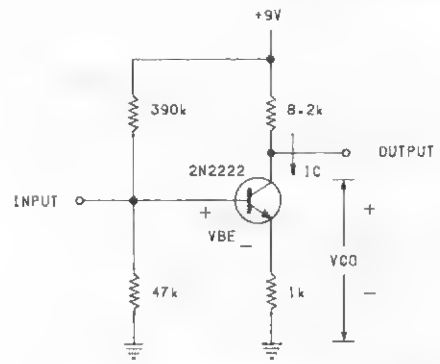


Fig. 7-12. Circuit for Example 7-4.

The program assumes that silicon transistors are used so that the forward biased base-emitter voltage VBE is 0.65 (in line 110). If any one of the five resistors is represented by an open circuit, as is frequently the case with the feedback resistor RF, then the value of 1E12 (10^{12}) should be used.

Example 7-4

For the circuit of Fig. 7-12, determine the 2N2222 transistor operating point assuming a typical beta of 200 and a leakage current I_{ceo} of 10 μA.

The computed results are shown in Fig. 7-13.

For designing a single stage common-emitter amplifier, as shown in Fig. 7-14, the TRANSAMP program of Fig. 7-15 is used. Having a particular transistor in mind, the program requires the following parameters:

1. Maximum collector-emitter voltage V_{CE}
2. Maximum collector current, in milliamperes
3. Maximum power dissipation, in milliwatts
4. Maximum junction temperature, in degrees Celsius
5. Leakage current, I_{co}, in microamperes
6. Input impedance, h_{ie}
7. Minimum, typical, and maximum beta (h_{FE})

It is assumed that silicon transistors are used. If germanium types are used then change the value of V3 in line 112 to 0.3.

In addition, the following design constraints are required:

1. Maximum ambient temperature, in degrees Celsius
2. Supply voltage
3. Quiescent collector current, in milliamperes

```
100 'TRANSISTOR BIASING (BIAS)
101 CLS:PRINT"          TRANSISTOR BIASING PARAMETERS"
102 INPUT"SUPPLY VOLTAGE ";V
103 INPUT"COLLECTOR RESISTOR RC ";RC
104 INPUT"EMITTER RESISTOR RE ";RE
105 INPUT"FEEDBACK RESISTOR RF ";RF
106 INPUT"BASE RESISTOR RA ";RA
107 INPUT"BASE RESISTOR RB ";RB
108 INPUT"TRANSISTOR BETA (HFE) ";B
109 INPUT"TRANSISTOR ICEO ";IO
110 V2=0.65
111 R1=RA*RB/(RA+RB)
112 V1=RB*V/(RA+RB)
113 R=RC+RF
114 RR=RE*R*IO/B+R1*(V-V2)
115 Q=R1*RE*IO/B+R*(V1-V2+IO*R1/B)
116 S=R1*RE*(1+1/B)+R1*R/B+RC*R1
117 T=RE*R*(1+1/B)
118 U=(Q+RR)/(S+T)
119 V3=V-RC*U
120 PRINT:PRINT"IC";U*1000;"MA":PRINT"IB";U/B*1E6;"UA":
    PRINT"IE";B/(B-1)*U*1000;"MA"
121 PRINT"VCO";V3;"VOLTS"
122 END
```

Fig. 7-11. Listing for BIAS program.

```
        TRANSISTOR BIASING PARAMETERS

SUPPLY VOLTAGE ? 9
COLLECTOR RESISTOR RC ? 8.2E3
EMITTER RESISTOR RE ? 1E3
FEEDBACK RESISTOR RF ? 1E12
BASE RESISTOR RA ? 390E3
BASE RESISTOR RB ? 47E3
TRANSISTOR BETA (HFE) ? 200
TRANSISTOR ICEO ? 10E-6


IC = .263525 MA
IB = 1.31763 UA
IE = .26485 MA
VCO = 6.83909 VOLTS
```

Fig. 7-13. Results for Example 7-4.

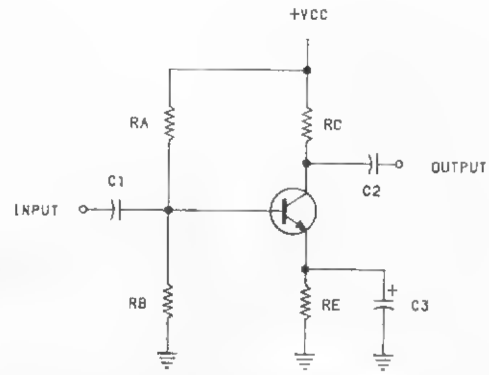


Fig. 7-14. Common-emitter amplifier circuit.

```
100 'TRANSISTOR COMMON EMITTER AMPLIFIER DESIGN (TRANSAMP)
101 CLS:PRINT,"COMMON EMITTER AMPLIFIER DESIGN"
102 PRINT
103 INPUT"MAXIMUM TRANSISTOR VCE ";V1
104 INPUT"MAXIMUM COLLECTOR CURRENT (IN MA) ";I1
105 INPUT"MAXIMUM POWER DISSIPATION IN MW ";P1
106 INPUT"MAXIMUM JUNCTION TEMP IN DEG. C ";T2
107 IF T2>0 THEN 109
108 INPUT"TRANSISTOR THERMAL RESISTANCE ";T1
109 INPUT"ICO IN UA ";I2
110 I2=I2/100
111 INPUT"HFE MIN, TYPICAL, MAX ";B1,B2,B3
112 V3=0.7
113 INPUT"SUPPLY VOLTAGE ";V2
114 INPUT"QUIESCENT COLLECTOR CURRENT IN MA ";I5
115 INPUT"%VCC ACROSS EMITTER RESISTOR ";V5
116 V5=V5*V2/100
117 INPUT"AMBIENT TEMPERATURE IN DEG. C ";T3
118 INPUT"MAX % CHANGE FOR IC ";I4
119 INPUT"HIE IN K-OHMS ";Z3
120 INPUT"LOW-FREQUENCY 3DB POINT IN HZ ";F1
121 IF V2<.9*V1 THEN 123
122 PRINT"----------SUPPLY VOLTAGE TOO HIGH----------"
123 IF I5<.4*I1 THEN 125
124 PRINT"----------COLLECTOR CURRENT TOO HIGH----------"
125 IF V2*I5<.8*P1 THEN 127
126 PRINT"----------POWER DISSIPATION TOO HIGH----------"
127 IF V2*I5<.5*P1 THEN 129
128 PRINT"----------USE A HEAT SINK----------"
129 IF T2<=0 THEN 131
130 T1=(T2-25)/P1
131 IF T3+V2*I5*T1<T2 THEN 133
132 PRINT"----------JUNCTION TEMP TOO HIGH----------"
133 R4=V5/I5
134 R3=(V2-V5)/(2*I5)
135 I3=I2*(2[((T3-25)/10)-1)
136 S1=I4*I5/(100*I3)
137 S3=(B3+1)*B1*S1*I3/(I5*(B3-B1))
138 S2=S1*S3/(S1+S3)
139 R5=(1+B2)*(S2-1)*R4/(B2+1-S2)
140 R1=R5*V2/(V5+V3)
141 R2=V2*R5/(V2-V5-V3)
142 Z1=R5*Z3/(R5+Z3)
```

Continued on next page.

Fig. 7-15. Listing for TRANSAMP program.

```
143 R4=R4*1000
144 PI=3.1415927
145 C1=5E3/(PI*F1*Z1)
146 C2=5E6/(PI*R4*F1)
147 C3=5E3/(PI*F1*R3)
148 A1=B2*R3/Z3
149 A2=A1*A1+Z3/R3
150 A3=10*LOG(A2)/LOG(10)
151 CLS:PRINT"DESIGN SPECS:"
152 PRINT"R1 =";INT(R1*100+.5)/100;"K-OHMS"
153 PRINT"R2 =";INT(R2*100+.5)/100;"K-OHMS"
154 PRINT"R3 =";INT(R3*100+.5)/100;"K-OHMS"
155 PRINT"R4 =";R4;"OHMS"
156 PRINT"C1 =";INT(C1*10+.5)/10;"UF"
157 PRINT"C2 =";INT(C2*10+.5)/10;"UF"
158 PRINT"C3 =";INT(C3*10+.5)/10;"UF"
159 V4=.35*(V2-V5)
160 V6=V4/A1
161 PRINT"AV =";A1
162 PRINT"AP =";A2;"      ";INT(A3*10+.5)/10;" DB"
163 PRINT"ZIN =";INT(Z1*100+.5)/100;"K-OHMS"
164 PRINT"ZOUT =";INT(R3*100+.5)/100;"K-OHMS"
165 PRINT"VIN MAX =";INT(V6*1000+.5)/1000
166 PRINT"VOUT MAX ";INT(V4*1000+.5)/1000
167 PRINT:INPUT"DO YOU WANT TO CHANGE PARAMETERS (YES/NO) ";ZZ$
168 IF ZZ$="YES" THEN 114
169 PRINT:END
```

Fig. 7-15 (cont). Listing for TRANSAMP program.

4. Percentage of the supply voltage across emitter resistor R4
5. Maximum percentage change for the collector current
6. Low-frequency 3-dB point

When executed, the program determines the value for the four resistors and three capacitors, the ac voltage and power gains, the input and output impedances, and the maximum input and output voltages of the amplifier.

Example 7-5

For the circuit of Fig. 7-14, design a common-emitter amplifier operating from a 9-volt supply with a minimum input frequency of 50 Hz. For a particular transistor, the parameters are:

Maximum V_{CE} = 45 volts
Maximum collector current = 30 mA
Maximum power dissipation = 300 mW
Maximum junction temperature = 175 degrees Celsius
I_{co} = 0.002 μA
Minimum beta = 60
Typical beta = 150
Maximum beta = 350
h_{ie} = 5 kilohms

In addition, the quiescent collector current is to be 2 mA, the collector current is to vary up to a maximum of 100%, and the maximum expected ambient temperature is 30 degrees centigrade.

The computed output results are shown in Fig. 7-16, giving the resultant circuit of Fig. 7-17.

HEAT SINK DESIGN

Selecting the proper heat sink is just as important as the design of transistor or integrated circuits. Using the thermal model shown in Fig. 7-18, the HEATSINK program of Fig. 7-19 determines the proper heat sink required when the following parameters are specified:

1. Power dissipation of the device
2. Maximum junction temperature
3. Maximum ambient temperature
4. Junction-case thermal resistance

The maximum junction temperature and junction-case thermal resistance are usually given in the data sheet for the particular device that you are using. In addition, the program requires the method in which the device is to be mounted to the heat sink, which is one of the following:

```
                COMMON EMITTER AMPLIFIER DESIGN

MAXIMUM TRANSISTOR VCE ? 45
MAXIMUM COLLECTOR CURRENT (IN MA) ? 30
MAXIMUM POWER DISSIPATION IN MW ? 300
MAXIMUM JUNCTION TEMP IN DEG. C ? 175
ICO IN UA ? .002
HFE MIN, TYPICAL, MAX ? 60,150,350
SUPPLY VOLTAGE ? 9
QUIESCENT COLLECTOR CURRENT IN MA ? 2
%VCC ACROSS EMITTER RESISTOR ? 10
AMBIENT TEMPERATURE IN DEG. C ? 30
MAX % CHANGE FOR IC ? 100
HIE IN K-OHMS ? 5
LOW-FREQUENCY 3DB POINT IN HZ ? 50

DESIGN SPECS:
R1 = 349.06 K-OHMS
R2 = 75.47 K-OHMS
R3 = 2.03 K-OHMS
R4 = 450 OHMS
C1 = 6.9 UF
C2 = 70.7 UF
C3 = 15.7 UF
AV = 60.75
AP = 3693.03       35.7  DB
ZIN = 4.63 K-OHMS
ZOUT = 2.03 K-OHMS
VIN MAX = .047
VOUT MAX  2.835

DO YOU WANT TO CHANGE PARAMETERS (YES/NO) ? NO

READY
>_
```

Fig. 7-16. Results for Example 7-5.

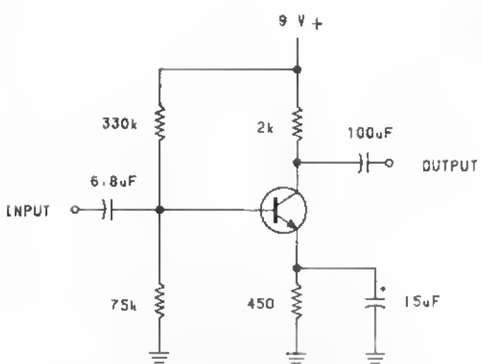


Fig. 7-17. Final circuit for Example 7-5.

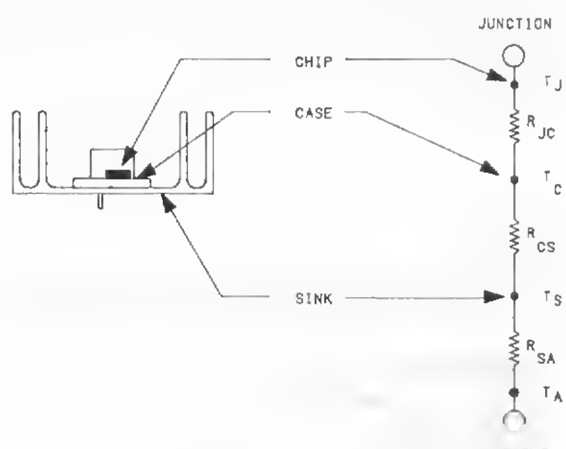


Fig. 7-18. Transistor heat sink, thermal model.

Dry
Grease
Dry with mica washer
Grease with mica washer

The program requires all temperatures to be expressed in Celsius, and determines the maximum thermal resistance of the required heat sink. As an alternative, the approximate volume of the heat sink may be calculated, using Fig. 7-20 as a guide for those who like to build their own, using aluminum.

```
100 'HEAT SINK DESIGN PROGRAM (HEATSINK)
101 CLS:PRINT,"HEAT SINK DESIGN"
102 PRINT:PRINT"ALL TEMPERATURES ARE TO BE EXPRESSED IN DEGREES CENTIGRADE"
103 PRINT:INPUT"REQUIRED POWER DISSIPATION (WATTS) ";P
104 INPUT"MAXIMUM AMBIENT TEMPERATURE ";TA
105 INPUT"MAXIMUM JUNCTION TEMPERATURE ";TJ
106 INPUT"JUNCTION-CASE THERMAL RESISTANCE (DEGREES/WATT) ";JC
107 PRINT:PRINT"SELECT TYPE OF MOUNTING:"
108 PRINT"      1. DRY"
109 PRINT"      2. GREASE"
110 PRINT"      3. DRY WITH MICA WASHER"
111 INPUT"      4. GREASE WITH MICA WASHER  ";M
112 ON M  GOTO 113  ,114  ,115  ,116
113 RC=0.25:T$="DRY":GOTO 117
114 RC=0.1:T$="GREASE":GOTO 117
115 RC=0.8:T$="DRY WITH MICA WASHER":GOTO 117
116 RC=0.4:T$="GREASE WITH MICA WASHER"
117 TS=TJ-P*(RC+JC)
118 RJ=(TJ-TA)/P
119 RS=RJ-(JC+RC)
120 TC=RC*P+TS:TJ=(RC+JC)*P+TS
121 IF TS>55 THEN 122  ELSE 125
122 TS=55:RS=(TS-TA)/P:TC=RC*P+TS:TJ=(RC+JC)*P+TS
123 PRINT:INPUT"DO YOU WANT TO KNOW THE APPROXIMATE SINK VOLUME REQUIRED ";A$
124 IF A$="YES" THEN 127
125 GOSUB 132
126 GOTO 130
127 V=46.155*RS↑(-1.52145):V=INT(V*100+.5)/100
128 GOSUB 132
129 PRINT"APPROXIMATE HEAT SINK VOLUME REQUIRED =";V;"CU IN"
130 PRINT"REQUIRED HEAT SINK THERMAL RESISTANCE =";RS;"C/WATT OR LESS"
131 PRINT:END
132 CLS:PRINTTAB(29)"RESULTS"
133 PRINT:PRINT"USING ";T$;" CASE-SINK MOUNTING:"
134 PRINT:PRINT"MAXIMUM AMBIENT TEMPERATURE =";TA;"C"
135 PRINT"MAXIMUM JUNCTION TEMPERATURE =";TJ;"C"
136 PRINT"HEAT SINK TEMPERATURE =";TS;"C"
137 PRINT"CASE-SINK TEMPERATURE =";TC;"C"
138 PRINT"POWER DISSIPATION =";P;"WATTS"
139 RETURN
```

Fig. 7-19. Listing for HEATSINK program.

Example 7-6

Determine the proper heat sink for a transistor that has the following electrical characteristics:

1. Dissipates 10 watts
2. Maximum junction temperature = 135 degrees Celsius
3. Junction-case resistance = 6.2 degrees/watt

In addition, the maximum expected ambient temperature is 30 degrees, and the transistor is to be mounted with a mica washer with grease.

As shown in Fig. 7-21, the heat sink should have a thermal resistance of 2.5 degrees/watt or less, and which should have an approximate minimum volume of 11.45 in^3.

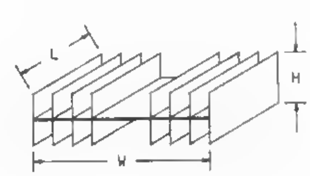


Fig. 7-20. Heat sink diagram.

```
               HEAT SINK DESIGN

ALL TEMPERATURES ARE TO BE EXPRESSED IN DEGREES CENTIGRADE

REQUIRED POWER DISSIPATION (WATTS) ? 10
MAXIMUM AMBIENT TEMPERATURE ? 30
MAXIMUM JUNCTION TEMPERATURE ? 135
JUNCTION-CASE THERMAL RESISTANCE (DEGREES/WATT) ? 6.2

SELECT TYPE OF MOUNTING:
     1. DRY
     2. GREASE
     3. DRY WITH MICA WASHER
     4. GREASE WITH MICA WASHER  ? 4

DO YOU WANT TO KNOW THE APPROXIMATE SINK VOLUME REQUIRED ? YES


         RESULTS

USING GREASE WITH MICA WASHER CASE-SINK MOUNTING:

MAXIMUM AMBIENT TEMPERATURE = 30 C
MAXIMUM JUNCTION TEMPERATURE = 121 C
HEAT SINK TEMPERATURE = 55 C
CASE-SINK TEMPERATURE = 59 C
POWER DISSIPATION = 10 WATTS
APPROXIMATE HEAT SINK VOLUME REQUIRED = 11.45 CU IN
REQUIRED HEAT SINK THERMAL RESISTANCE = 2.5 C/WATT OR LESS


READY
>_
```

Fig. 7-21. Results for Example 7-6.

8

Additional Routines

This final chapter presents several mathematical routines which aid the solution of a number of circuit problems. Included are the four-quadrant arctangent function, mathematical operations on complex numbers, and the minimum and maximum and maximum value of an array.

FOUR-QUADRANT ARCTANGENT FUNCTION

Although Level II contains the arctangent function ATN (Z), it is only valid for those arguments between $-\pi/2$ (−90 degrees) and $+\pi/2$ (+90 degrees). As an example, if the argument Z is expressed as the ratio of Y and X coordinates, such that $Z = Y/X$, then ATN (Y/X) will yield correct results as long as X is greater than zero (ie., first or fourth quadrant). If X is negative, ATN (Y/X) will still return a result that is between −1.57080 and +1.57080, which is in error, since the point actually lies in either the second or third quadrant.

The ARCTAN subroutine listed in Fig. 8-1 determines the four-quadrant arctangent of the two variables X and Y and returns the answer expressed either in degrees or radians using the variables AD and AR, respectively, as shown by the simple main program of Fig. 8-2. As will be discussed in the next section, the ARCTAN subroutine is necessary when converting rectangular coordinate, or complex numbers to their exponential or polar equivalents.

Example 8-1

Using the main program of Fig. 8-2, determine the arctangent in degrees and radians for the following X-Y coordinates:

X	Y
2	5
2	−5
−2	5
−2	−5

```
6000 ' 4-QUADRANT ARCTANGENT SUBROUTINE (ARCTAN)
6001 ' INPUTS: X AND Y
6002 ' OUTPUTS: AR = ARCTAN IN RADIANS; AD = ARCTAN IN DEGREES
6003 IF Y=0 THEN 6004 ELSE 6007
6004 IF X<0 THEN 6005 ELSE 6006
6005 AR=-3.1415927:GOTO 6017
6006 AR=0:GOTO 6017
6007 IF X>0 THEN 6008 ELSE 6010
6008 IF Y<>0 THEN 6009 ELSE 6003
6009 AR=ATN(Y/X):GOTO 6017
6010 IF X=0 THEN 6011 ELSE 6014
6011 IF Y<0 THEN 6012 ELSE 6013
6012 AR=-1.5707963:GOTO 6017
6013 AR=1.5707963:GOTO 6017
6014 IF X<0 THEN 6015 ELSE 6017
6015 IF Y<>0 THEN 6016 ELSE 6003
6016 AR=3.1415927+ATN(Y/X)
6017 AD=AR*57.2957795:RETURN
```

Fig. 8-1. Listing for ARCTAN subroutine.

```
100 'MAIN PROGRAM FOR ARCTAN
101 CLS:INPUT"VALUE FOR X ";X
102 INPUT"VALUE FOR Y ";Y
103 GOSUB 6000             'ARCTAN SUBROUTINE
104 PRINT"THEN ATN(Y/X) :"
105 PRINT"      ";AR;"RADIANS"
106 PRINT"      ";AD;"DEGREES"
107 PRINT:INPUT"HIT <ENTER> TO CONTINUE ";Z9
108 GOTO 101
109 END
```

Fig. 8-2. Simple main program for ARCTAN.

Fig. 8-3 gives the computer results.

COMPLEX NUMBER MATH

The ability to perform arithmetic operations with complex numbers in either rectangular or polar form can make any resulting computations easier. The CPLXMATH program shown in Fig. 8-4 will:

1. Convert a number from rectangular to polar form
2. Convert a number from polar to rectangular form
3. Add two complex numbers
4. Subtract two complex numbers
5. Multiply two complex numbers
6. Divide one complex number by another

```
VALUE FOR X ? 2
VALUE FOR Y ? 5
THEN ATN(Y/X) :
       1.19029 RADIANS
       68.1986 DEGREES

HIT <ENTER> TO CONTINUE ?

VALUE FOR X ? 2
VALUE FOR Y ? -5
THEN ATN(Y/X) :
      -1.19029 RADIANS
      -68.1986 DEGREES

HIT <ENTER> TO CONTINUE ?

VALUE FOR X ? -2
VALUE FOR Y ? 5
THEN ATN(Y/X) :
       1.9513 RADIANS
       111.801 DEGREES

HIT <ENTER> TO CONTINUE ?

VALUE FOR X ? -2
VALUE FOR Y ? -5
THEN ATN(Y/X) :
       4.33188 RADIANS
       248.199 DEGREES

HIT <ENTER> TO CONTINUE ?
```

Fig. 8-3. Results for Example 8-1.

By choosing either addition, subtraction, multiplication, or division, the program allows both numbers to be expressed in any combination of forms. In addition, the resulting answer can be chosen to be expressed in rectangular, or polar form, or both.

Example 8-2

Convert the complex number $5 - j3$ to its equivalent polar form.

As shown in Fig. 8-5, the equivalent polar form has a magnitude of 5.831 and an angle of −31 degrees.

Example 8-3

Convert to rectangular form the polar number that has a magnitude of 6.92 and an angle of 56.1 degrees.

As shown in Fig. 8-6, the equivalent complex number in rectangular form is $3.86 + j5.744$.

Example 8-4

Divide the number "A", $5 - j3$, by the number "B", having a magnitude of 6.92 and an angle of 56.1 degrees, and express the answer in both rectangular and polar form.

As shown in Fig. 8-7, we enter the number "3" when asked to select the forms of the two numbers, as A and B are in rectangular and polar forms, respectively.

MINIMUM AND MAXIMUM VALUES OF AN ARRAY

The MINMAX subroutine of Fig. 8-8 determines the minimum and maximum values of the array X(I). This routine is used in many of the plotting routines of Chapter 2, the DAMPED pro-

```
100 ' COMPLEX ARITHMETIC PROGRAM (CPLXMATH)
101 CLEAR
102 T=57.2957795
103 CLS:PRINT"CHOOSE MATHEMATICAL OPERATION TO BE PERFORMED:"
104 PRINT:PRINT"1. RECTANGULAR TO POLAR FOR A SINGLE NUMBER"
105 PRINT"2. POLAR TO RECTANGULAR FOR A SINGLE NUMBER"
106 PRINT"3. ADDITION OF TWO NUMBERS (A + B)"
107 PRINT"4. SUBTRACTION OF TWO NUMBERS (A - B)"
108 PRINT"5. MULTIPLICATION OF TWO NUMBERS (A*B)"
109 INPUT"6. DIVISION OF TWO NUMBERS  (A/B)      CHOICE ";J
110 IF J<3 THEN 123  ELSE 111
111 CLS:PRINT"SELECT INPUT FORMS FOR THE TWO VARIABLES"
112 PRINT:PRINT,"A","B"
113 PRINT"1.","RECTANGULAR","RECTANGULAR"
114 PRINT"2.","POLAR","POLAR"
115 PRINT"3.","RECTANGULAR","POLAR"
116 PRINT"4.","POLAR","RECTANGULAR";
117 INPUT"     ";I
118 IF J<3 THEN 123  ELSE 119
119 CLS:PRINT"PRINT THE ANSWER IN:"
120 PRINT:PRINT"1. RECTANGULAR FORM"
121 PRINT"2. POLAR FORM"
122 INPUT"3. BOTH FORMS    ";K
123 CLS:PRINT"ENTER:":GOSUB 175
124 IF J=1 THEN I=1
125 IF J=2 THEN I=2
126 ON I GOTO 127  ,128  ,129  ,130
127 AR=A1:AI=A2:BR=B1:BI=B2:GOTO 131
128 AM=A1:AA=A2:BM=B1:BA=B2:GOTO 131
129 AR=A1:AI=A2:BM=B1:BA=B2:GOTO 131
130 AM=A1:AA=A2:BR=B1:BI=B2
131 ON I GOTO 132  , 138  , 135  , 132
132 BM=SQR(BR↑2+BI↑2):Y=BI:X=BR:GOSUB 187
133 BA=ZD
134 IF I=4 GOTO 139
135 AM=SQR(AR↑2+AI↑2):Y=AI:X=AR:GOSUB 187
136 AA=ZD
137 IF I<>3 GOTO 140
138 BR=BM*COS(BA/T):BI=BM*SIN(BA/T)
139 AR=AM*COS(AA/T):AI=AM*SIN(AA/T)
140 ON J GOTO 145  ,145  ,141  ,142  ,143  ,144
141 CR=AR+BR:CI=AI+BI:GOTO 145
142 CR=AR-BR:CI=AI-BI:GOTO 145
143 CM=AM*BM:CA=AA+BA:GOTO 145
144 CM=AM/BM:CA=AA-BA
145 ON J GOTO 151  ,151  ,146  ,146  ,149  ,149
146 ON K GOTO 151  ,147  ,147
147 CM=SQR(CR↑2+CI↑2):X=CR:Y=CI:GOSUB 187
148 CA=ZD
149 ON K GOTO 150  , 154  , 150
150 CR=CM*COS(CA/T):CI=CM*SIN(CA/T)
151 AR=INT(AR*1000+.5)/1000:AI=INT(AI*1000+.5)/1000
152 CR=INT(CR*1000+.5)/1000:CI=INT(CI*1000+.5)/1000
153 AM=INT(AM*1000+.5)/1000:AA=INT(AA*10+.5)/10
154 CM=INT(CM*1000+.5)/1000:CA=INT(CA*10+.5)/10
155 ON J GOTO 156  ,158  ,160  ,166  ,167  ,173
156 PRINT:PRINT"THEN:":PRINT
157 PRINTAR;"+ J(";AI;") = ";AM;"/ (";AA;")":GOTO 174
158 PRINT:PRINT"THEN:":PRINT
159 PRINTAM;"/ (";AA;") = ";AR;"+ J(";AI;")":GOTO 174
160 C$="+":PRINT:PRINT"THEN:":PRINT:IF K-2 <0 THEN 161  ELSE 162
```

Continued on next page.

Fig. 8-4. Listing for CPLXMATH program.

```
161 PRINT"A ";C$;" B = ";CR;"+ J(";CI;")":GOTO 174
162 IF K-2=0 THEN 163  ELSE 164
163 PRINT"A ";C$;" B = ";CM;"/ (";CA;")":GOTO 174
164 PRINT"A ";C$;" B = ";CR;"+ J(";CI;")"
165 PRINT"       = ";CM;"/ (";CA;")":GOTO 174
166 C$="-":PRINT:PRINT"THEN:":PRINT:IF K-2<0 THEN 161  ELSE 162
167 C$="*":PRINT:PRINT"THEN:":PRINT:IF K-2<0 THEN 168  ELSE 169
168 PRINT"A ";C$;" B = ";CR;" + J(";CI;")":GOTO 174
169 IF K-2=0 THEN 170  ELSE 171
170 PRINT"A ";C$;" B = ";CM;"/ (";CA;")":GOTO 174
171 PRINT"A ";C$;" B = ";CR;" + J(";CI;")"
172 PRINT"       = ";CM;"/ (";CA;")":GOTO 174
173 C$="/":PRINT:PRINT"THEN:":PRINT:IF K-2<0 THEN 168  ELSE 169
174 PRINT:PRINT:END
175 'INPUT SUBROUTINE
176 IF J=1 THEN 179  ELSE 177
177 IF J=2 THEN 182  ELSE 178
178 IF I=1 OR I=3 THEN 179  ELSE 182
179 PRINT:INPUT"REAL PART (A) ";A1:INPUT"IMAGINARY PART (A) ";A2
180 IF J=1 OR J=2 THEN 181  ELSE 184
181 RETURN
182 PRINT:INPUT"MAGNITUDE (A) ";A1:INPUT"ANGLE (A) ";A2
183 IF J=1 OR J=2 THEN RETURN
184 IF I=1 OR I=4 THEN 185  ELSE 186
185 INPUT"REAL PART (B) ";B1:INPUT"IMAGINARY PART (B) ";B2:RETURN
186 INPUT"MAGNITUDE (B) ";B1:INPUT"ANGLE (B) ";B2:RETURN
187 ' 4-QUADRANT ARCTANGENT SUBROUTINE (ARCTAN)
188 IF ABS(Y)<1E-4 THEN Y=0
189 IF ABS(X)<1E-4 THEN X=0
190 AU=Y/X
191 IF ABS(AU)<1E-4 THEN AU=0
192 IF Y=0 THEN 193  ELSE 196
193 IF X<0 THEN 194  ELSE 195
194 ZR=-3.1415927:GOTO 206
195 ZR=0:GOTO 206
196 IF X>0 THEN 197  ELSE 199
197 IF Y<>0 THEN 198  ELSE 192
198 ZR=ATN(AU):GOTO 206
199 IF X=0 THEN 200  ELSE 203
200 IF Y<0 THEN 201  ELSE 202
201 ZR=-1.5707963:GOTO 206
202 ZR=1.5707963:GOTO 206
203 IF X<0 THEN 204  ELSE 206
204 IF Y<>0 THEN 205  ELSE 192
205 ZR=3.1415927+ATN(AU)
206 ZD=ZR*57.2957795:RETURN
```

Fig. 8-4 (cont). Listing for CPLXMATH program.

```
CHOOSE MATHEMATICAL OPERATION TO BE PERFORMED:

1. RECTANGULAR TO POLAR FOR A SINGLE NUMBER
2. POLAR TO RECTANGULAR FOR A SINGLE NUMBER
3. ADDITION OF TWO NUMBERS (A + B)
4. SUBTRACTION OF TWO NUMBERS (A - B)
5. MULTIPLICATION OF TWO NUMBERS (A*B)
6. DIVISION OF TWO NUMBERS  (A/B)     ? 1

ENTER:

REAL PART (A) ? 5
IMAGINARY PART (A) ? -3

THEN:

 5 + J(-3 ) =  5.831 / (-31 )

READY
>_
```

Fig. 8-5. Results for Example 8-2.

```
CHOOSE MATHEMATICAL OPERATION TO BE PERFORMED:

1. RECTANGULAR TO POLAR FOR A SINGLE NUMBER
2. POLAR TO RECTANGULAR FOR A SINGLE NUMBER
3. ADDITION OF TWO NUMBERS (A + B)
4. SUBTRACTION OF TWO NUMBERS (A - B)
5. MULTIPLICATION OF TWO NUMBERS (A*B)
6. DIVISION OF TWO NUMBERS  (A/B)     ? 2

ENTER:

MAGNITUDE (A) ? 6.92
ANGLE (A) ? 56.1

THEN:

 6.92 / ( 56.1 ) =  3.86 + J( 5.744 )

READY
>_
```

Fig. 8-6. Results for Example 8-3.

```
CHOOSE MATHEMATICAL OPERATION TO BE PERFORMED:

1. RECTANGULAR TO POLAR FOR A SINGLE NUMBER
2. POLAR TO RECTANGULAR FOR A SINGLE NUMBER
3. ADDITION OF TWO NUMBERS (A + B)
4. SUBTRACTION OF TWO NUMBERS (A - B)
5. MULTIPLICATION OF TWO NUMBERS (A*B)
6. DIVISION OF TWO NUMBERS  (A/B)     ? 6

SELECT INPUT FORMS FOR THE TWO VARIABLES

                  A              B
1.                RECTANGULAR    RECTANGULAR
2.                POLAR          POLAR
3.                RECTANGULAR    POLAR
4.                POLAR          RECTANGULAR     ? 3
```

Fig. 8-7. Results for Example 8-4.

```
PRINT THE ANSWER IN:

1. RECTANGULAR FORM
2. POLAR FORM
3. BOTH FORMS     ? 3

ENTER:
REAL PART (A) ? 5
IMAGINARY PART (A) ? -3
MAGNITUDE (B) ? 6.92
ANGLE (B) ? 56.1

THEN:

A / B =  .043   + J(-.842 )
      =  .843 / (-87.1 )

READY
>_
```

Fig. 8-7 (cont). Results for Example 8-4.

gram of Chapter 3, and the DSTATS program of Chapter 4. The required input variables are the array and its length N. When control returns to the main program, the minimum and maximum values are found in the variables X1 and X2, respectively.

Example 8-5

Using the simple main program of Fig. 8-9, determine the minimum and maximum values of the values listed in Table 2-1.

The resulting output is shown in Fig. 8-10.

```
7000 'MINIMUM AND MAXIMUM VALUES OF AN ARRAY (MINMAX)
7001 'X(I) = INPUT POINTS
7002 'N = LENGTH OF THE ARRAY
7003 'X1 = MINIMUM VALUE    X2 = MAXIMUM VALUE
7004 X1=X(1):X2=X(1)
7005 FOR I=2 TO N
7006 IF (X1-X(I))<=0 THEN 7008 ELSE 7007
7007 X1=X(I):GOTO 7010
7008 IF (X2-X(I))<0 THEN 7009 ELSE 7010
7009 X2=X(I)
7010 NEXT I
7011 RETURN
```

Fig. 8-8. Listing for MINMAX subroutine.

```
100 'MAIN PROGRAM FOR MINMAX
101 DIM X(100)
102 CLS:INPUT"LENGTH OF THE ARRAY ";N
103 FOR I=1 TO N
104 PRINT"VALUE #";I;
105 INPUT X(I)
106 NEXT I
107 GOSUB 7000              'MINMAX SUBROUTINE
108 PRINT:PRINT"MINIMUM VALUE = ";X1
109 PRINT"MAXIMUM VALUE = ";X2
110 END
```

Fig. 8-9. Simple main program for MINMAX.

```
LENGTH OF THE ARRAY ? 14
VALUE # 1 ? 92
VALUE # 2 ? 94
VALUE # 3 ? 96
VALUE # 4 ? 99
VALUE # 5 ? 101
VALUE # 6 ? 104
VALUE # 7 ? 110
VALUE # 8 ? 90
VALUE # 9 ? 93
VALUE # 10 ? 95
VALUE # 11 ? 98
VALUE # 12 ? 100
VALUE # 13 ? 103
VALUE # 14 ? 106


MINIMUM VALUE =  90
MAXIMUM VALUE =  110

READY
>_
```

Fig. 8-10. Results for Example 8-5.

Appendix **A**

Programs and Subroutines

This appendix lists all the programs and subroutines discussed in this book, along with the approximate memory required (in bytes).

Name	*Memory*	*Description*
ARCTAN	600	Four-quadrant arctangent function
ATTNPAD1	1400	Resistive attenuator pads without graphics
ATTNPAD2	2600	Resistive attenuator pads with graphics
BIAS	700	Analysis of generalized bias network
BNDPASS1	2800	Bandpass filter design for Q less than 10
BNDPASS2	2400	Bandpass filter design for Q less than 50
CPLXMATH	3400	Complex number mathematics
CPLXSEQ	4700	Solution of simultaneous equations with complex coefficients
DAMPED	3500	Damped oscillation waveform analysis
DFTRANS	7300	Discrete fast Fourier transform
DSTATS	1500	Descriptive statistics of a sample
FSERIES	3800	Fourier series
HEATSINK	1500	Heatsink selection and design
ILAPLACE	2200	Inverse Laplace transform of a transfer function
LCPAD1	700	Lossless L-C pads without graphics
LCPAD2	1200	Lossless L-C pads with graphics
LCPSPAD1	1700	Lossless L-C pads-phase shift without graphics
LCPSPAD2	2800	Lossless L-C pads-phase shift with graphics
LINEPLOT	2100	Cartesian plotting of a continuous function—line printer
LOGLOG	3400	Log-Log plotting—video display
LPHP	9800	Design of low- and high-pass filters, second through sixth order
MINMAX	1000	Minimum and maximum value of an array
MLREGRES	6100	Multiple linear regression
NOTCH	2700	Design of notch filters
PCARTXY	1600	Cartesian plotting—video display
PHISTGM	1600	Histogram—video display
PLYROOTS	2900	Real and complex roots of a polynomial
POLARPLT	2400	Polar coordinate plotting—line printer
PSEMILOG	2500	Semilogarithmic plotting—video display
REGRESSN	9900	Regression between two variables

RMSAV	2600	RMS and average value for discrete data points or function
SIMULEQ	3100	Solution of simultaneous equations with real coefficients
SVFILTER	5200	Design of state-variable filters
TEEPI	1600	Tee-Pi (Y-Delta) transformation
TIMER555	3200	555 timer monostable and astable multivibrators
TRANSAMP	1800	Design of common-emitter amplifier
TUNEDBP	3500	Design of staggered-tuned bandpass filters
VPOLAR	2100	Polar coordinate plotting—video display
XLOGPLOT	1800	Semilogarithmic plotting—line printer
ZENER	700	Zener diode regulator design

Appendix **B**

Standard Resistor and Capacitor Values

RESISTORS

1. The following ±5% standard decade values are available. Those marked with * are the ones most readily available from electronic suppliers.

1.0*	1.8*	3.3*	5.6*
1.1	2.0	3.6	6.2
1.2*	2.2*	3.9*	6.8*
1.3	2.4	4.3	7.5
1.5*	2.7*	4.7*	8.2*
1.6	3.0	5.1	9.1

To obtain standard resistance values, multiply preferred number from decade table by powers of 10. Standard values are available from 10Ω to 22 MΩ.

2. The following ±1% values are available, but at a higher cost.

10.0	12.1	14.7	17.8	21.5	26.1	31.6	38.3	46.4	56.2	68.1	82.5
10.2	12.4	15.0	18.2	22.1	26.7	32.4	39.2	47.5	57.6	69.8	84.5
10.5	12.7	15.4	18.7	22.6	27.4	33.2	40.2	48.7	59.0	71.5	86.6
10.7	13.0	15.8	19.1	23.2	28.0	34.0	41.2	49.9	60.4	73.2	88.7
11.0	13.3	16.2	19.6	23.7	28.7	34.8	42.2	51.1	61.9	75.0	90.0
11.3	13.7	16.5	20.0	24.3	29.4	35.7	43.2	52.3	63.4	76.8	93.1
11.5	14.0	16.9	20.5	24.9	30.1	36.5	44.2	53.6	64.9	78.7	95.3
11.8	14.3	17.4	21.0	25.5	30.9	37.4	45.3	54.9	66.5	80.6	97.6

These standard values are available from 10Ω to 22.1 MΩ.

CAPACITORS

In general, capacitor values follow the standard decade values for ±10% resistors. The values listed below are in microfarads, and those marked with * are the ones most readily available from electronic suppliers.

.001*	.0033*	.01*	.033*	.1*	.33*
.0012	.0039	.012	.039	.12	.39
.0015	.0047*	.015	.047*	.15	.47*
.0018	.005	.018	.05	.18	.5
.002	.0056	.02	.056	.2	.56
.0022*	.0068*	.022*	.068*	.22*	.68*
.0025	.0075	.025	.075	.25	.75
.0027	.0082	.027	.082	.27	.82

Index

A

Active filters, 15
Ambient temperature, 49
Analysis, damped oscillations, 41-42
Aperiodic waveforms, 33
Arctangent function, four-quadrant, 125
Arithmetic mean, 58
Array (s)
  maximum values, 126, 130
  minimum values, 126, 130
  variable, 13
Astable, 114
Asymmetry, 58
Average values of waveform, 34-36
Axes, 9

B

Bandpass
  filter, 99, 103
    staggered-tuned Butterworth, 107
Bar graph, 9
Bessel, 87
Beta, 116
Bipolar transistor, 116
Bode plots, 15
Butterworth
  bandpass filters, staggered-tuned, 107
  polynomials, 63

C

Cartesian plots, 11-14, 19
Cascading, 110
Center frequency, 110
Chebyshev polynomials, 63
Circuit, multivibrator, 114
Complex number math, 126
Continuous mathematical functions, 12
Cosine, 36
Current (s)
  gain, 49
  leakage, 116
  loop, 75
Curves, least squares regression, 49, 50, 56, 58

D

Damped oscillations, analysis, 41-42
Data points, discrete, 12
Delta-Wye transform, 75
Design, 555 timer, 114
Diagram, scatter, 49
Discrete data points, 12
Duty cycle, 114

E

Exponential, 50

F

Filter (s)
  active, 15
  bandpass, 99, 103
  high-pass, 87
  low-pass, 87
  notch, 108-110
  passive, 15
  Sallen-Key, 87
  second-order, low-pass, 19
  staggered-tuned Butterworth bandpass, 107
  state-variable, 87, 96, 99, 107
  wideband, 110
555 timer design, 114
Four-quadrant arctangent function, 125
Fourier, 14
  series, 36-39
  transform, 39-41
Frequency, 49
  center, 110
Function (s)
  continuous mathematical, 12
  four-quadrant arctangent, 125

G

Cain, passband, 96
Ceometric, 50
Craph, 9

H

H-pads, 70
Heat sink design, 120-122
High-pass filters, 87
Histograms, 9-11

I

Impedance, 49
  matching pads, 68-75
Inverse, 50
  Laplace transform, 63-68

J

Junction temperature, 122

K

Kirchhoff's
  current law, 81
  voltage law, 81
Kurtosis, 58

L

Laplace transform, inverse, 63-68
Leakage current, 116
Least squares regression curves, 49, 50, 56, 58
Line
  printer
    plotting with, 19, 21
    using, 7
Log-log plots, 17
Loop currents, 75
Lossless pads, 73-75
Low-pass filters, 19, 87

M

Magnitude, 15
  Fourier series, 36
Matching pads, impedance, 68-75
Math, complex number, 126
Mathematical functions, continuous, 12
Maximum values of array, 126, 130
Merging program with subroutine, 8
Mesh current network analysis, 75-83
Minimum values of array, 126, 130
Monostable, 114
Multivibrator circuit, 114

N

Network analysis, mesh current, 75-83
Nonrepetitive waveforms, 33
Notch filter, 108-110
Nyquist plots, 17

O

O-pads, 70
Oscillator, voltage controlled, 41
Output voltage, quiescent, 116

P

Pads
  impedance matching, 68-75
  lossless, 73-75
  resistance, 70
Passband gain, 96
Passive filters, 15
Periodic waveforms, 33
Phase
  angle, 36
  Fourier series, 36
  -locked loop, 41
  responses, 15
Pi
  -networks, 70
  -tee transform, 75
Plot(s)
  Bode, 15
  cartesian, 11-14
  driver program, 11
  log-log, 17
  Nyquist, 17
  polar, 17, 19, 21
  semilogarithmic, 14-17
Plotting with line printer, 19, 21
Points, discrete data, 12
Polar plots, 17, 19, 21
Polynomial(s), 50
  Butterworth, 63
  Chebyshev, 63
  roots, 63

Q

Quiescent output voltage, 116

R

Radians, 14
Regression curves, least squares, 49, 50, 56, 58
REGRESSN program, 49
Relationship between variables, 49, 50, 56, 58
Resistance
  pads, 70
  thermal, 121
Rms values of waveform, 34-36
Roots of polynomials, 63

S

Sallen-Key filter, 87
Sample statistics, 58
Scatter diagram, 49
Second-order
  function, 15
  low-pass filter, 19
Semilogarithmic plots, 14-17, 19
Simpson's rule, 34
Skewness, 58
Spectrum plot, 41
Staggered-tuned Butterworth bandpass filters, 107
Standard deviation, 58
State-variable filter, 87, 96, 99, 107
Statistics, sample, 58
Subroutine, merging with program, 8

T

T-networks, 70
Temperature
  ambient, 49
  junction, 122
Thermal resistance, 121
Timer design, 555, 114
Triangle waves, 39
Transform
  Delta-Wye, 75
  inverse Laplace, 63-68
  Pi-Tee, 75
Transient response, 12
Transistor(s), 116-120
  operating point, 49

U

Unbalanced pads, 70
Using line printer, 7

V

Variable(s)
  arrays, 13
  relationship between, 49, 50, 56, 58
Variance, 58
Voltage
  controlled oscillator, 41
  quiescent output, 116
  regulator design, zener diode, 113

W

Waveform(s)
  aperiodic, 33
  average values, 34-36
  nonrepetitive, 33
  periodic, 33
  rms values, 34-36
Waves, triangle, 39
Wideband filter, 110

Z

Z-scores, 58
Zener diode voltage regulator design, 113

CIRCUIT DESIGN PROGRAMS FOR THE TRS-80®

There are a number of relatively inexpensive "personal-type" digital computers, such as Radio Shack's Model TRS-80, which can efficiently perform virtually any type of computation required for the design and analysis of electronic networks. Thus, this book presents a variety of useful BASIC language programs that will greatly simplify the design and analysis of commonly encountered circuit problems.

A variety of programs are presented that enable the user to solve a myriad of problems, such as rms and average values, periodic waveform, inverse Laplace, Fourier, design of matching pads, attenuators, active filters, heat sinks, integrated circuit timer, zener diode regulator, and bipolar transistor circuits.

All output is displayed on the video screen, but most programs can be easily modified so that the results are handled by a line printer, cassette tape, or disk.

Appendix A lists all the programs discussed in this book along with their approximate memory capacities. Appendix B lists the standard values for resistors and capacitors, which are helpful for a number of the programs.

This book is for the reader who has a good understanding of Level II BASIC. It was written primarily for the advanced computer user.

Sams Software Programs include 37 of the technical programs contained in this book. These tested, debugged, documented programs are available on cassettes and can be used as stand-alone programs or as subroutines for more complex programs. See page 140 of this book for ordering information and more details about Sams Technical Software.

Howard M. Berlin is an electrical engineer with the Chemical Systems Laboratory at Aberdeen Proving Ground, Maryland. He has been an adjunct instructor in the Department of Electrical Engineering at the University of Delaware. He has been primarily involved in biomedical engineering research and physiological instrumentation. He has taught short courses for the Department of the Army, graduate courses at the University of Delaware, and short courses at several universities and conferences. He is the author of a number of governmental reports and articles for scientific and amateur radio magazines. In addition, he is the author of the following books: **The 555 Timer Applications Sourcebook, With Experiments; Design of Active Filters, With Experiments; Design of Op-Amp Circuits, With Experiments; Design of Phase-Locked Loop Circuits, With Experiments; Guide to CMOS Basics, Circuits, & Experiments;** and coauthor with Robert T. Stone of **Design of VMOS Circuits, With Experiments.** He is a member of Sigma Xi, IEEE, and the Delaware Academy of Medicine. He is an active ham radio operator (W3HB).

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

ISBN: 0-672-21741-4